ELSEVIER

Artificial Intelligence 106 (1998) 77–107

Artificial Intelligence

2U: an exact interval propagation algorithm for polytrees with binary variables

Enrico Fagiuoli [a,1], Marco Zaffalon [b,*]

[a] *Università degli Studi di Milano, Dipartimento di Matematica, via Cicognara 7, 20129 Milano, Italy*
[b] *IDSIA, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, c.so Elvezia 36, 6900 Lugano, Switzerland*

Received 28 May 1997; received in revised form 19 March 1998

Abstract

This paper addresses the problem of computing posterior probabilities in a discrete Bayesian network where the conditional distributions of the model belong to convex sets. The computation on a *general* Bayesian network with convex sets of conditional distributions is formalized as a global optimization problem. It is shown that such a problem can be reduced to a combinatorial problem, suitable to exact algorithmic solutions. An exact propagation algorithm for the updating of a polytree with binary variables is derived. The overall complexity is linear to the size of the network, when the maximum number of parents is fixed. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Bayesian networks; Convex sets; Credal sets; Intervals; Uncertain reasoning; Inference

1. Introduction

Bayesian networks have been revealed as one of the main tools in domains where uncertain reasoning is needed [12].

In the discrete case, the model requires the specification of a certain number of probabilities to be estimated from data or from existing knowledge about the domain. Very often, the assumption of availability of such probabilities is not realistic. Many factors like economic and temporal constraints, ignorance about the phenomenon and group decision problems may partially inhibit the above estimate [6,8]. A step further can be made by taking into account the partial knowledge of the distribution. One way of achieving this result is to adopt convex sets of probability distributions generated by linear constraints

* Corresponding author. Email: zaffalon@idsia.ch.
[1] Email: fagiuoli@milano.ccr.it.

0004-3702/98/$ – see front matter
PII: S0004-3702(98)00089-7

[2,11,13,15] (or polytopes of distributions). In the Bayesian network case, this means that every prior or posterior distribution on a node can be defined as belonging to a polytope.

Polytopes of distributions, also called *credal sets* [8] and equivalent to *coherent lower previsions* [16], are a very general tool to formalize partial probabilistic knowledge. Their flexibility is given by the possibility of specifying probabilistic information via linear constraints on the distribution, i.e., by characterizing the uncertainty with a set of possible distributions, where none of them is preferable to another: the credal set formalizes the ignorance on the phenomenon (see the recent work of Walley [16] for an introduction to the subject and for a comparison with classical probability theory, the Dempster–Shafer theory and with fuzzy logic). The advantages of working with constraints on the distribution are clear: the partial ignorance about the domain is included in the model, hence the user is not forced, explicitly or implicitly, to insert a subjective point of view in order to fix a single distribution. Of course, the less knowledge, the less precision is given by inferences. In fact, given that the distribution belongs to a set, every probability value belongs to an interval, whose extremes are the minimum and the maximum of the probability value when the distribution varies in the credal set. If the ignorance is greater, the interval is wider, thus providing the user with less information. But the (exact) interval allows the user of the system to study the domain under every possible condition [8,9] given his/her state of knowledge.

However, there are disadvantages related to the treatment of credal sets over Bayesian networks, for structural and complexity reasons that mix. Structurally, credal sets cannot simply be seen as a straight generalization of classical (point) probability. There is a number of basic issues of probability theory that must be rediscussed with credal sets. For instance, operations like Bayes theorem or the chain rule, have no equivalent counterpart, nor a concept analogous to probabilistic independence exists [1,4,7]. The ability to decompose the computation into smaller pieces is fundamental for every propagation scheme. Hence propagation algorithms for Bayesian networks are hard to build, credal sets seem to resist propagation [6]: since a propagation algorithm cannot rely on the basic probability operations above, the decomposition of the main problem into subproblems is completely demanded to the algorithm, whose conceptual complexity increases.

Credal sets are very close to the optimization world [17], because their use always implies the computation of the extremes of some quantity. Also from such a point of view, the problems raised by credal sets over Bayesian networks are difficult (see Section 4.1). In a way, this helps understanding the reasons why from the early attempts to realize the propagation on Bayesian networks [11,13], there has not been a substantial jump towards their effective treatment in significant cases. This issue is closely related to the computational complexity of working with credal sets [3]. In literature, the first exact approach to the propagation of polytopes of distributions on a network structure is given by Cano et al. [5]. The authors derive an exact algorithm of propagation that is based on the manipulation of *extreme* probabilities, i.e., distributions corresponding to the vertices of the set of distributions. Unfortunately, this pure combinatorial way of treating credal sets is not viable, because it is subject to a combinatorial explosion of the number of extreme probabilities to manipulate. A different approach is given by Chrisman [6] who defines a new conditioning rule and provides an exact general algorithm for Markovian-like 2-monotone lower probabilities (a less expressive tool as compared to coherent lower

previsions [16]) on a junction tree. In this case the requirement of 2-monotonicity seems quite a stringent global property together with Markovianity. Furthermore, in the discrete case, there is still a relevant complexity problem that forces the application to junction trees with *small* cliques. The complexity of dealing with credal sets suggests to study more closely the problem, and if it is the case, to limit the study to significant subcases, that, if solved, can also serve to start the development of new methodologies.

This paper focuses on a precise definition of the problem of inference in Bayesian networks with credal sets. This is made on the basis of optimization. The optimization view of inference in a general Bayesian network defined with credal sets is characterized in a formal way. This allows the combinatorial nature of such problems to be shown (Section 4.1). Then, the attention is restricted to the case of the singly-connected Bayesian networks with binary random variables. In this case, credal sets coincide with intervals, i.e., any probability value defining the model belongs to an interval. For the above set of Bayesian networks, an efficient inference process is shown to be possible. The first linear-time algorithm (called 2-Updating: 2U) for a wide significant set of Bayesian networks is derived. The derivation methodology is a mixture of analytical decompositions of the global problem and of combinatorial solution of the smaller problems generated by the first part. The algorithm exhibits a scheme similar to Pearl's belief updating. It uses a message flow to update the node probability values, where nodes are interpreted as processors connected with communication links (arcs). The message flow is the same as in the original updating, but both the extreme values of a message are passed and the rules of composition are different. Within the class of networks with a fixed maximum number of parents, the time complexity is $O(L)$ where L is the size of the maximum path of the net. The computations local to a node (which determine the coefficient of the linear form above) depend on a $O(2^{2n_{\max}})$ term, where $n_{\max}$ is the maximum number of parents for a node in the graph. In the original belief updating, applied to a polytree with binary variables, the latter term is $O(2^{n_{\max}})$. This result is such that 2U enables credal sets over large Bayesian networks to be easily investigated, which is appealing both for applications and for research.

The paper is structured in the following way. In Section 2, some basic definitions and notations used in the sequel are provided. In Section 3 a point-probability version of 2U is developed. In Section 4, the optimization problems to be dealt with for treating credal sets over Bayesian networks are characterized; then the propagation formulas derived in Section 3 are extended to the interval case. Section 5 discusses the computational complexity of the resulting algorithm and Section 6 presents a full numerical example of application of 2U. The concluding section discusses the relevance of the results and the issues to be addressed for the definition of more general algorithms. The Appendix A contains the proofs of the theorems stated in Section 4.

2. Definitions and notations

The following conventions are used. Any opportune set, say $\Im$, is used as a set of indexes for some variables. In such a case, a single element of $\Im$ corresponds to a variable, and every $\Im' \subseteq \Im$ corresponds to a vector of variables.

Now, some basic tools for the definition of a Convex Bayesian Network are provided. They comprise the graph, the set of random variables and the set of conditional probability variables.

- Let $G = (N, A)$ be a directed acyclic graph with N equal to the set of the nodes and $A \subseteq N \times N$ the set of arcs. $\forall i \in N$ define the set of its parents as $Pa(i) = \{j \in N \mid (j, i) \in A\}$.
- Random variables are indexed by the elements of N. $\forall i \in N$, denote with X_i a random variable with values in Ω_i, $|\Omega_i| < \infty$; $\forall W \subseteq N$ denote with X_W the vector of random variables with values in $\times_{j \in W} \Omega_j$. If K and W are nonempty sets of nodes such that $K \subseteq W$, let $X_K^{\downarrow W}$ denote the vector X_K obtained from X_W by dropping the variables related to $W \setminus K$.
- Conditional probability variables are real-valued variables indexed by the following set of triples, $\{(i, X_i, X_{Pa(i)}) \in N \times \Omega_i \times \Omega_{Pa(i)}\}$. They are denoted by $\varsigma_{i,X_i}^{X_{Pa(i)}}$. $\forall i \in N, \forall X_i \in \Omega_i$ and $\forall X_{Pa(i)} \in \Omega_{Pa(i)}$, $\varsigma_{i,X_i}^{X_{Pa(i)}}$ stands for the probability $P[X_i \mid X_{Pa(i)}]$. The conditional distribution $P[\cdot \mid X_{Pa(i)}]$ is then represented by the real-valued $|\Omega_i|$-dimensional vector indexed by $\{(i, X_i, X_{Pa(i)}) \mid X_i \in \Omega_i\}$, and is denoted for short with $\varsigma_i^{X_{Pa(i)}}$.

Now, only consider interval constraints on the conditional probabilities of the model; the general case is analogous. The interval to which a generic probability (ς) belongs, is denoted by $[\underline{\varsigma}, \overline{\varsigma}]$. By the notations above, it is clear that a set of intervals for the generic conditional distribution $P[\cdot \mid X_{Pa(i)}]$ of the model is

$$\left\{\underline{P}[X_i \mid X_{Pa(i)}] \leqslant P[X_i \mid X_{Pa(i)}] \leqslant \overline{P}[X_i \mid X_{Pa(i)}],\ X_i \in \Omega_i\right\}.$$

This is equivalent to defining the following convex set (a polytope) of real-valued vectors

$$\wp_i^{X_{Pa(i)}} = \left\{ \varsigma_i^{X_{Pa(i)}} \in \mathbb{R}^{|\Omega_i|} \;\middle|\; \underline{\varsigma}_{i,X_i}^{X_{Pa(i)}} \leqslant \varsigma_{i,X_i}^{X_{Pa(i)}} \leqslant \overline{\varsigma}_{i,X_i}^{X_{Pa(i)}} \right.$$

$$\left. \forall X_i \in \Omega_i,\ \sum_{X_i \in \Omega_i} \varsigma_{i,X_i}^{X_{Pa(i)}} = 1 \right\}.$$

The definition of Interval Bayesian Network is based on the sets $\wp_i^{X_{Pa(i)}}$.

Definition 1. Let $(G, \wp)$ be a pair such that G is defined as above and

$$\wp = \left\{ \varsigma \in \mathbb{R}^{|\Omega_N|} \;\middle|\; \varsigma = [\varsigma_{X_N}]_{X_N \in \Omega_N},\ \varsigma_{X_N} = \prod_{i \in N} \varsigma_{i,X_i^{\downarrow N}}^{X_{Pa(i)}^{\downarrow N}},\ \varsigma_i^{X_{Pa(i)}^{\downarrow N}} \in \wp_i^{X_{Pa(i)}^{\downarrow N}} \right\}$$

where $\wp \neq \emptyset$. This is called an *Interval Bayesian Network.*

$\wp$ is the set of all the joint distributions obtained by making every possible choice of the conditional probabilities in the sets $\wp_i^{X_{Pa(i)}}$. Therefore an interval Bayesian network represents a set of Bayesian networks. This definition can naturally be extended to the case where the sets $\wp_i^{X_{Pa(i)}}$ are built by means of general linear constraints, not only bounds on

the probabilities. In that case, the sets $\wp_i^{X_{Pa(i)}}$ are general probability polytopes, and the model can be referred to as *Convex Bayesian Network*, or *Credal Network*, for short.

In the following sections, a set of nodes and the corresponding vector of random variables are denoted with the same symbol when no ambiguity can arise. Furthermore, when dealing with random variables, for any variable $X \in N$, the two elements of Ω_X are denoted with x and $\bar{x}$ (recall that the symbol X stands for the generic random variable. In the sequel, when X is used in a formula, it means that both x and $\bar{x}$ can substitute X. In the case when x is used, the formula is developed for x only). Finally, a quantity whose value depends on the chosen distribution in $\wp$ is referred to with the terminology "probability variable".

3. Design of 2U for point probabilities

In this paper, the propagation of intervals of probability is based on the extension of a point-probability updating algorithm defined here. In Section 3.1, Pearl's belief updating [12] is briefly recalled. On this basis, Section 3.2 discusses the need of a different updating algorithm. Finally, in Section 3.3 the specialization of the belief updating to the case of binary variables allows the new algorithm to be derived.

3.1. Review of Pearl's updating

Consider the fragment of an $|N|$-nodes singly-connected network shown in Fig. 1.

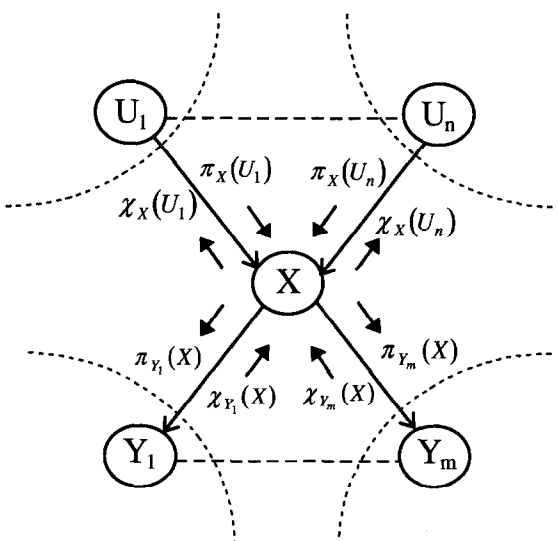

Fig. 1. A fragment of a singly connected net.

The net is supposed to contain the evidence $E = e$, which is a set of nodes whose random variables have an observed state that by definition has the property $P[E = e] \neq 0$. Define $U = \{U_1, \ldots, U_n\}$ as the set of parents of a generic node X.

The quantity $P[X \mid e]$ must be computed for every node X in the network. Pearl shows the correctness of the following propagation formulas [12]:

$$P[X \mid e] = \alpha\pi(X)\lambda(X), \tag{1}$$

$$\pi(X) = \sum_{U} P[X \mid U] \prod_{i} \pi_X(U_i), \tag{2}$$

$$\lambda(X) = \prod_{j} \lambda_{Y_j}(X), \tag{3}$$

$$\pi_{Y_j}(X) = \pi(X) \prod_{k \neq j} \lambda_{Y_k}(X), \tag{4}$$

$$\lambda_X(U_i) = \beta \sum_{X} \lambda(X) \sum_{U_k \neq U_i} P[X \mid U] \prod_{k \neq i} \pi_X(U_k) \tag{5}$$

where α and β are constants that are not taken into account during the computation. In this case, the product $\pi(X)\lambda(X)$ in Eq. (1) gives $P[X, e]$ [12]. Then $P[X \mid e]$ is simply computed by normalizing $P[X, e]$, $P[X \mid e] = P[X, e] / \sum_X P[X, e]$. The quantities $\pi(X)$ and $\lambda(X)$ are calculated with (2) and (3) exploiting the messages $\pi_X(U_i)$ and $\lambda_{Y_j}(X)$, respectively, coming from the parents and the children of X. Formulas (4) and (5) define the messages that X, respectively, sends to its children and to its parents and are needed in order to update the rest of the network about the current state of X. Finally, there exist three particular cases to be dealt with, the source nodes, the barren nodes and the evidence nodes. Pearl shows that for a source node, $\pi(X) = P[X]$ and for a barren node, $\lambda(X) = 1$. If X is an evidence node, say $X = x$, then it is supposed to have a *dummy* child, $\widetilde{Y}$, that sends the message $\lambda_{\widetilde{Y}}(X = x) = 1$ and $\lambda_{\widetilde{Y}}(X) = 0 \; \forall X \neq x$.

3.2. *Reasons behind the new updating*

Pearl's propagation formulas are developed for any singly-connected network defined with point-probabilities. As they are, it seems difficult to extend them to the propagation of credal sets. The main reason for this is that formulas (1)–(5) do not allow the posterior probability $P[X \mid e]$ to be computed in a straightforward way. In fact, the computation is made by omitting the constants α and β and hence obtaining the joint probability $P[X, e]$ in the place of $P[X \mid e]$. The latter is produced only after the normalization of $P[X, e]$. In the credal set case, every probability has a minimum and a maximum value. A straight extension of the above formulas might lead to the extremes of $P[X, e]$ and of $P[e]$. But the computation of the extreme values for $P[X \mid e] = P[X, e]/P[e]$ cannot be realized in general by simply knowing the minimum and the maximum value of $P[X, e]$ and of $P[e]$ separately. Indeed these two quantities are not independent,[2] because they generally

[2] Notice that this is a different concept as compared to the independence of random variables.

belong to a certain definition set. Hence the extension to credal sets is based on a different updating algorithm, for which the constants α and β are absorbed into the new formulas and then $P[X \mid e]$ is computed by means of the composition of terms that are independent. Under this condition of independence, knowing the extremes of the above terms is enough in order to compute the extremes of $P[X \mid e]$. An analogous argument holds for any other quantity that is treated by the propagation algorithm during the computation. If such a quantity can be built by means of independent pieces, the extension to the computation of its extremes is easy to realize.

3.3. Derivation of the new updating

The aim of the present section is to define the new updating for point probabilities that is based on the principle of independence cited above. This is obtained by creating the algorithm in such a way that all the information that a node receives along an arc is restricted to one real number. In other words, a node receives just one number along any of its arcs, and such numbers are necessarily independent, since they come from disjoint subnetworks.

The new updating is derived on the basis of Eqs. (1)–(5), which are, in part, reformulated taking advantage of the assumption of dealing with binary variables. Such a new algorithm is extended to the intervals in Section 4.

3.3.1. Computation of $P[x \mid e]$

First, notice that it is sufficient to only take the case $X = x$ into account, since $P[\bar{x} \mid e] = 1 - P[x \mid e]$. Eq. (1) implies that,

$$
\begin{aligned}
P[x \mid e] &= \frac{P[x \mid e]}{P[x \mid e] + P[\bar{x} \mid e]} \\
&= \frac{\alpha\pi(x)\lambda(x)}{\alpha\pi(x)\lambda(x) + \alpha\pi(\bar{x})\lambda(\bar{x})} && (6) \\
&= \left(1 + \left(\frac{1}{\pi(x)} - 1\right)\frac{\lambda(\bar{x})}{\lambda(x)}\right)^{-1} && (7) \\
&= \left(1 + \left(\frac{1}{\pi(x)} - 1\right)\frac{1}{\Lambda^X}\right)^{-1} && (8)
\end{aligned}
$$

where the new quantity $\Lambda^X = \lambda(x)/\lambda(\bar{x})$ is defined.

Notice that formula (8) does not consider some particular cases that must be then treated apart. Such cases are generated because the values $\pi(x)$, $\lambda(x)$ and $\lambda(\bar{x})$ are probabilities [12] that can also be zero, and for which the formula cannot be applied. Therefore, an extension of formula (8) that is shown to produce the right results is provided here. First of all, observe that $\alpha\pi(x)\lambda(x) + \alpha\pi(\bar{x})\lambda(\bar{x}) = 1$, and for this reason $\lambda(x)$ and $\lambda(\bar{x})$ cannot be both zero, i.e., it may only happen that Λ^X is 0 or $1/0$. For the same reason, when $\pi(x) = 0$, Λ^X cannot be $1/0$. Hence, the critical cases are reduced to two: one or both between $\pi(x)$ and Λ^X are 0; Λ^X is $1/0$. Consider the first case. It happens when $\pi(x)\lambda(x) = 0$. Formula (6) shows that this implies $P[x \mid e] = 0$. Now notice that taking the limit of expression (8) when $\pi(x)\lambda(x) \to 0$, brings to the same result. For example,

consider the case when $\pi(x) \to 0$ and $\lambda(x) \to 0$ (hence $\Lambda^X \to 0$). Expression (8) becomes $(1 + (\infty - 1)\infty)^{-1}$, i.e., 0. Here the value 0 is obtained as the result of a limit operation. But it can also be interpreted as the result of formula (8) when it is extended to treat infinite values. Such an extension is simply realized by adopting the symbol ∞ and the usual associated algebra (like, for example, $1/0 = \infty$, $1/\infty = 0$, etc.) for manipulating the expressions that contain it. This interpretation is possible if no indeterminacy can arise (like $0 \cdot \infty$, for instance); this is the case of the current formula and also of the formulas in the next sections. In other words, it is sufficient to permit the use of the ∞ in order to make formula (8) treat the special cases homogeneously with respect to the others, and the reason for this is just the proof above. This is also valid for the second case, namely, when Λ^X is $1/0$ (i.e., $\lambda(\bar{x}) = 0$). In fact, the value of $P[x \mid e]$ in such a case is 1, as shown by Eq. (6), and the same value is obtained by letting Λ^X be ∞ in Eq. (8).

3.3.2. Computation of $\pi_{Y_j}(x)$

$\pi_{Y_j}(x)$ is derived in a completely analogous way to what done with regards to $P[x \mid e]$ in Section 3.3.1. In fact, $\pi_{Y_j}(x)$ is equal to $P[x \mid e]$ if the evidence in the subnetwork with root Y_j is suppressed [12]. For this reason, it is enough to repeat the derivation above, without taking into account the information carried by the message $\lambda_{Y_j}(X)$, i.e., using $\prod_{k \neq j} \lambda_{Y_k}(X)$ in the place of $\lambda(X) = \prod_k \lambda_{Y_k}(X)$.

3.3.3. Computation of Λ^X

The new formula for Λ^X is achieved by means of Eq. (3),

$$\Lambda^X = \frac{\lambda(x)}{\lambda(\bar{x})} = \frac{\prod_j \lambda_{Y_j}(x)}{\prod_j \lambda_{Y_j}(\bar{x})} = \prod_j \left(\frac{\lambda_{Y_j}(x)}{\lambda_{Y_j}(\bar{x})} \right) = \prod_j \Lambda^X_{Y_j} \tag{9}$$

where $\Lambda^X_{Y_j}$ is interpreted as a message that Y_j sends to X. Observe that it can happen that $\lambda(\bar{x}) = 0$ (and in that case $\lambda(x)$ cannot be zero, as discussed above), and hence the domain of Λ^X must be extended to the ∞ in order to include the cited case.

3.3.4. Computation of $\Lambda^X_{Y_j}$

Recall that $\Lambda^X_{Y_j} = \lambda_{Y_j}(x)/\lambda_{Y_j}(\bar{x})$, according to the definition given in Section 3.3.3. First of all, observe that the messages $\lambda_{Y_j}(x)$ and $\lambda_{Y_j}(\bar{x})$ cannot be both zero, otherwise formula (3) would bring to $\lambda(x) = \lambda(\bar{x}) = 0$, which is impossible, as shown in Section 3.3.1.

Now $\Lambda^X_{Y_j}$ is redefined on the basis of formula (5). Denote the probability

$$\sum_{U_k \neq U_i} P[X \mid U] \prod_{k \neq i} \pi_X(U_k)$$

by $\rho[X \mid U_i]$. It follows that,

$$\Lambda^{U_i}_X = \frac{\beta \sum_X \lambda(X) \rho[X \mid u_i]}{\beta \sum_X \lambda(X) \rho[X \mid \bar{u}_i]} = \frac{\lambda(x)\rho[x \mid u_i] + \lambda(\bar{x})(1 - \rho[x \mid u_i])}{\lambda(x)\rho[x \mid \bar{u}_i] + \lambda(\bar{x})(1 - \rho[x \mid \bar{u}_i])}. \tag{10}$$

Dividing by $\lambda(\bar{x})$ (for the moment consider only the case $\lambda(\bar{x}) \neq 0$), Eq. (10) becomes,

$$\Lambda_X^{U_i} = \frac{\Lambda^X \rho[x \mid u_i] + (1 - \rho[x \mid u_i])}{\Lambda^X \rho[x \mid \bar{u}_i] + (1 - \rho[x \mid \bar{u}_i])} \tag{11}$$

$$= \frac{\rho[x \mid u_i] + (\Lambda^X - 1)^{-1}}{\rho[x \mid \bar{u}_i] + (\Lambda^X - 1)^{-1}} \tag{12}$$

dividing Eq. (11) by $\Lambda^X - 1$ and only considering the case $\Lambda^X \neq 1$.

The two excluded cases can be treated by the same formula, if it is extended in order to treat the infinite. This can be seen more clearly by rewriting formula (12) as follows,

$$\Lambda_X^{U_i} = \frac{\rho[x \mid u_i]}{\rho[x \mid \bar{u}_i] + (\Lambda^X - 1)^{-1}} + \frac{1}{\frac{\rho[x \mid \bar{u}_i]}{(\Lambda^X - 1)^{-1}} + 1}. \tag{13}$$

Consider the first case, namely $\lambda(\bar{x}) = 0$. In this case $\Lambda_{Y_j}^X = \rho[x \mid u_i]/\rho[x \mid \bar{u}_i]$, as shown by Eq. (10). The same result is obtained as follows. When $\lambda(\bar{x}) = 0$, Λ^X is ∞; using $\Lambda^X = \infty$ in Eq. (13), in the same way introduced in Section 3.3.1, it is exactly $\Lambda_{Y_j}^X = \rho[x \mid u_i]/\rho[x \mid \bar{u}_i]$. The second case corresponds to $\Lambda^X = 1$. This happens because $\lambda(x) = \lambda(\bar{x})$; in this case, Eq. (10) gives $\Lambda_X^{U_i} = 1$. Observe that the same result is obtained by Eq. (13), using infinite values.

3.3.5. *Summary of the propagation formulas*

The propagation formulas derived in the previous sections are listed below:

$$P[x \mid e] = \left(1 + \left(\frac{1}{\pi(x)} - 1\right)\frac{1}{\Lambda^X}\right)^{-1}, \tag{14}$$

$$\pi(x) = \sum_U P[x \mid U] \prod_i \pi_X(U_i), \tag{15}$$

$$\Lambda^X = \prod_j \Lambda_{Y_j}^X, \tag{16}$$

$$\pi_{Y_j}(x) = \left(1 + \left(\frac{1}{\pi(x)} - 1\right)\frac{1}{\prod_{k \neq j} \Lambda_{Y_k}^X}\right)^{-1}, \tag{17}$$

$$\Lambda_X^{U_i} = \frac{\rho[x \mid u_i] + (\Lambda^X - 1)^{-1}}{\rho[x \mid \bar{u}_i] + (\Lambda^X - 1)^{-1}} \tag{18}$$

where

$$\rho[X \mid U_i] = \sum_{U_k \neq U_i} P[X \mid U] \prod_{k \neq i} \pi_X(U_k). \tag{19}$$

The three particular cases of the original updating are treated by a straightforward application of the messages definitions. For a source node, it is $\pi(x) = \pi_{Y_j}(x) = P[x]$, while a barren node sends the message $\Lambda^X = \Lambda_{U_i}^X = 1$. If X is an evidence node, it is supposed to have a dummy child, $\tilde{Y}$, that sends the message $\Lambda_{\tilde{Y}}^X = \infty$ if $X = x$ or $\Lambda_{\tilde{Y}}^X = 0$ if $X = \bar{x}$.

The message flow is exactly the same as in the original updating and for this reason the algorithm terminates in a time $O(L)$ where L is the length of the maximum path in the graph. The heaviest computation local to a node is related to $\pi(x)$ and, as in the belief updating, it depends on a $O(2^{n_{\max}})$ term. Hence the complexity is the same as in Pearl's updating.

4. Interval propagation by 2U

Formulas (14)–(19) are extended to propagate intervals. The optimization point of view is adopted for this purpose. The computation of the interval for a posterior probability of interest is equivalent to solving the following nonlinear, constrained, global optimization problems:

$$\underline{P}[x \mid e] = \min_{P[N] \in \wp} P[x \mid e], \qquad \overline{P}[x \mid e] = \max_{P[N] \in \wp} P[x \mid e]. \tag{20}$$

Observe that the domain is nonlinear since $\wp$ is defined via nonlinear constraints (Section 2) and that the objective function can be expressed as the ratio $P[x, e]/P[e]$, i.e., a ratio of linear functions. Problems (20) can equivalently be rewritten as the *global* optimization of a ratio of polynomials over a polytope (Theorem 7). Under this form, it is clear that also from a pure optimization point of view, such problems are difficult [14]. Furthermore, the number of dimensions of the space can be of the order of thousands (corresponding to the number of probability values in a Bayesian network), and hence the number of local optima can be huge.

The only key to solving problems of type (20) is exploiting its structure. In the present case, the graph of the Bayesian net can help structure the problem. By using the independence properties synthesized by the graph, the exact solution algorithm is developed as a network distributed optimization. That is, the extremes of the probability are built up by comparing the extremes of the messages involved in the computation; the extremes of the messages are created in the same way by considering other messages extreme values.

The development of the algorithm follows two steps.

First, some properties characterizing credal networks in the general case (i.e., not limited to single connection and binary variables) are derived. Specifically, it is defined a class of functions generated by credal networks (functions of type-3, in Definition 4) with the property of having extremes at the vertices of their linear definition set (Theorem 5). Such functions represent the general form of many common quantities that can be derived by a credal net. Therefore, they can be used as a general tool to prove the combinatorial nature of many problems generated by a credal net. In particular, they are used in this paper for solving all the optimization problems that are formulated in the derivation of 2U in the interval case (a more general example of application of Theorem 5 is given by Theorem 7, which shows that problem (20) can be rewritten as the optimization of type-3 functions, hence highlighting its combinatorial nature; notice that this is not directly connected to the derivation of 2U in the interval case).

The second step is the derivation of the propagation formulas, which is realized by extending formulas (14)–(18) to the intervals, by writing them as optimization problems that can be solved by means of the results of Section 4.1.

4.1. Probability functions in a credal network

This section formalizes a class of nonlinear constrained global optimization problems with the characteristic of having extremes at the vertices of the feasible set. This result allows the combinatorial nature of the underlying problem to be exploited in order to achieve an exact algorithmic solution, since the search of global optima can be restricted to a finite number of points.

Every optimization problem described in the formal derivation of 2U in Section 4.2 belongs to the above set and as such, the derivation is based on the theoretical result of Theorem 5. Moreover, Theorem 5 is a general result that is true for the most common quantities computed by a credal network. The optimization problem describing such a quantity can be shown to be within the hypotheses of Theorem 5.

The following notations are used.

Let $x_{\Im} \in \mathbb{R}^{|\Im|}$ be a vector of variables indexed by the set $\Im$. If $f : \Omega \subseteq \mathbb{R}^{|\Im|} \to \mathbb{R}$ is a function and $\Im' \subseteq \Im, \Im' \neq \emptyset$, it is denoted by $f_{\Im \setminus \Im'} : \Omega' \subseteq \mathbb{R}^{|\Im'|} \to \mathbb{R}$ any function that is obtained from f by fixing the values of the variables indexed by $\Im \setminus \Im'$.

Definition 2. If $f : \Omega \subseteq \mathbb{R}^{|\Im|} \to \mathbb{R}$ is a polynomial where every variable has degree strictly lower than 2, f is called a *type-1 function.*

Definition 3. If $f : \Omega \subseteq \mathbb{R}^{|\Im|} \to \mathbb{R}$ is defined as the ratio of two type-1 functions for which the denominator is never zero in Ω, then f is said *type-2 function.*

Definition 4. If $f : \Omega \subseteq \mathbb{R}^{|\Im|} \to \mathbb{R}$ is a function of type-1 or of type-2 and if there exists a partition of $\Im$ into the sets $\Im_1, \ldots, \Im_k$, such that $\forall j = 1, \ldots, k$, $f_{\Im \setminus \Im_j}$ is linear (if type-1) or is the ratio of two linear functions (if type-2), then f is a *type-3 function.*

Theorem 5. *Let $f : \Omega \subseteq \mathbb{R}^{|\Im|} \to \mathbb{R}$ be a type-3 function where the partition of $\Im$ is $\{\Im_1, \ldots, \Im_k\}$. Let the domain of $x_{\Im_j}$ $\forall j = 1, \ldots, k$ be denoted by Ω_j and let Ω_j be a closed polytope. If Ω is the closed $|\Im|$-dimensional polytope obtained when the vectors $x_{\Im_j}$ vary in their respective definition sets, then the extremes of f are in the set of vertices of Ω.*

Observe that functions of type 1, 2 and 3 are defined without resorting to a probabilistic interpretation, in order to give a general result, and for the same reason, the result provided by Theorem 5 does not depend on probability. But many quantities that can be obtained by a credal net can be mapped to such functions. Some example are: a prior probability value can be mapped to a type-1 function; a posterior probability to a type-2 function; type-3 functions can be used for the purpose of evaluating the extremes of a prior or a posterior probability over the particular domains that are generated by credal networks. Such a map exists for the above examples as shown by Theorem 7; Theorem 7 is also the formal rewriting of problem (20). Notice that Theorem 7 is not needed for the derivation of 2U. The derivation of 2U is based on Corollary 6, that is the specialization of Theorem 5 to the case of hyper-rectangular domains.

Corollary 6. *Let $f : \Omega \subseteq \mathbb{R}^{|\Im|} \to \mathbb{R}$ be a type-1 or a type-2 function, where the feasible set is described only with (closed) interval constraints on the variables. Then the function is within the hypotheses of Theorem 5 and hence its extreme points are in the set of vertices of Ω.*

Proof. Via the trivial proof of letting every set of the partition contain exactly one variable. □

The following theorem shows that the computation of the extremes of any prior or posterior probability of a general Bayesian network is a problem within the hypotheses of Theorem 5.

Theorem 7. *Consider the generic joint distribution $P \in \wp$ represented by a credal network. Let ξ be any (joint or marginal) prior or posterior probability of P. The optimization of ξ when $P \in \wp$, can be formulated as a problem satisfying the hypotheses of Theorem 5.*

4.2. Derivation of the algorithm

The following subsections present the derivation of the formulas for the interval propagation case. Three main observations are the following:

- Recall that when a global optimization problems is transformed into a combinatorial optimization problem, this is theoretically justified by the application of Corollary 6.
- The computation of probabilities in presence of intervals produces the problem of undefined probabilities. In fact, if the left value of a probability interval is zero, then it may be the case that for a certain distribution $P \in \wp$ and a set of nodes W, $P[W] = 0$. If $W \subseteq E$ was true, then $P[E] = 0$ and some of the propagation messages would be undefined. The above cases are formally taken into account by a limit consideration. The probability $P[X \mid e]$ can be evaluated in the limit of $P[W] \to 0$. For this reason in obtaining the formula, two steps are made: solving the optimization problem for a region that does not contain probability values equal to zero, and extending the solution to the general case by taking its limit when the extremes tend to the general feasible region. This process is illustrated in Section 4.2.1 where the complete procedure is undertaken. In the other sections the intermediate step is left implicit.
- Finally, there is an observation about the specification of the interval constraints on probabilities. It is observed that if a random variable, say $X \in \{x, \bar{x}\}$, is binary, then only one interval constraint is required to specify the (interval) distribution, because the remaining constraint is consequently determined. For instance, by giving $\underline{P}(x) \leqslant P(x) \leqslant \overline{P}(x)$, the implicit constraint $P(x) + P(\bar{x}) = 1$, this allows the constraint for $P(\bar{x})$, $1 - \overline{P}(x) \leqslant P(\bar{x}) \leqslant 1 - \underline{P}(x)$, to be obtained. This is not restrictive; had the model builder specified both the interval, it would always be possible to modify the constraints, in order to make them satisfy the above requirement, without altering the underlying set of distributions. The generalization of this property is referred to in literature as the reachability property of probability intervals [2]. For this reason, in the rest of the paper only the necessary interval constraints are considered.

4.2.1. Extremes of $P[x \mid e]$

Let us consider the minimum problem (the computation of the maximum is always analogous) by using Eq. (14) and observing that the feasible region of the problem can be described by bounds alone, because the probability variables $\pi(x)$ and Λ^X are independent. This is because they are provided by disjoint parts of the network (in the rest of the paper, the independence of other probability variables, when cited, is due to the same reason). The optimization problem is like follows,

$$\underline{P}[x \mid e] = \min_{\substack{a_\pi \leqslant \pi(x) \leqslant b_\pi \\ a_\Lambda \leqslant \Lambda^X \leqslant b_\Lambda}} P[x \mid e] = \min_{\substack{a_\pi \leqslant \pi(x) \leqslant b_\pi \\ a_\Lambda \leqslant \Lambda^X \leqslant b_\Lambda}} \left(1 + \left(\frac{1}{\pi(x)} - 1\right)\frac{1}{\Lambda^X}\right)^{-1},$$

where it is supposed $\underline{\pi}(x) \neq \overline{\pi}(x)$ and $\underline{\Lambda}^X \neq \overline{\Lambda}^X$ (because the opposite, point-probability cases are already treated in Section 3.3.1, also for degenerate values) and $a_\pi > \underline{\pi}(x)$, $b_\pi < \overline{\pi}(x)$, $a_\Lambda > \underline{\Lambda}^X$, $b_\Lambda < \overline{\Lambda}^X$, thus avoiding the problems due to possible degenerate probabilities (the latter hypotheses are relaxed by means of the passage to the limit below). Under these conditions, Corollary 6 can be applied, hence the solution can be looked for among the vertices of the rectangular region. It is easily noticed that the optimum is attained at $\pi(x) = a_\pi$ and $\Lambda^X = a_\Lambda$. The passage to the limit is made in such a way to enlarge the generic rectangle to the definition set, which can also be defined by means of intervals containing degenerate probability values (0 or 1),

$$\underline{P}[x \mid e] = \lim_{a_\pi \to \underline{\pi}(x),\ a_\Lambda \to \underline{\Lambda}^X} \left(1 + \left(\frac{1}{a_\pi} - 1\right)\frac{1}{a_\Lambda}\right)^{-1}.$$

This limit can always be solved simply by substitution since no indeterminacy arises. In fact it undergoes the same considerations developed in Section 3.3.1.

4.2.2. Extremes of $\pi(x)$

Recalling Eq. (15) and observing that the probability variables $P[x \mid U]$, $\pi_X(u_1), \ldots, \pi_X(u_n)$ are independent, the problem of minimum is formalized as follows,

$$\begin{aligned}
&\min \sum_U P[x \mid U] \prod_i \pi_X(U_i), \\
&\underline{P}[x \mid U] \leqslant P[x \mid U] \leqslant \overline{P}[x \mid U], && U \in \bigtimes_{i=1}^{n} \Omega_{U_i}, \\
&\underline{\pi}_X(u_i) \leqslant \pi_X(u_i) \leqslant \overline{\pi}_X(u_i), && i = 1, \ldots, n, \\
&\sum_{U_i \in \{u_i, \bar{u}_i\}} \pi_X(U_i) = 1, && i = 1, \ldots, n.
\end{aligned}$$

The last constraints can be removed from the formulation by letting, for instance, $\pi_X(\bar{u}_i) = 1 - \pi_X(u_i)$, and applying the substitution into the type-1 objective function. The new objective function is still a type-1 function and the new formulation satisfies the hypotheses of Corollary 6 since the new region is a hyper-rectangle. The problem can therefore be transformed into an equivalent combinatorial optimization one. The value of $P[x \mid U]$ at the optimum is easily fixed. Since $\prod_i \pi_X(U_i) \geqslant 0$ by definition, the optimum is achieved when $P[x \mid U] = \underline{P}[x \mid U]$. An exhaustive search on the states of $\pi_X(u_1), \ldots, \pi_X(u_n)$ completes the solution.

4.2.3. Extremes of Λ^X

The minimization is applied to Eq. (16),

$$\min \Lambda^X = \min_{\underline{\Lambda}^X_{Y_j} \leqslant \Lambda^X_{Y_j} \leqslant \overline{\Lambda}^X_{Y_j}} \prod_j \Lambda^X_{Y_j}.$$

The problem is such that Corollary 6 can be applied, and the solution is an immediate consequence. The minimum of Λ^X is obtained when all the $\Lambda^X_{Y_j}$ assume their minimum, that is

$$\underline{\Lambda}^X = \prod_j \underline{\Lambda}^X_{Y_j}.$$

4.2.4. Extremes of $\pi_{Y_j}(x)$

As noticed in Section 3.3.2, $\pi_{Y_j}(x)$ has a meaning analogous to $P[x \mid e]$. For this reason, the procedure for the computation of its minimum is completely analogous to the scheme adopted in Section 4.2.1, resulting in the following formula,

$$\underline{\pi}_{Y_j}(x) = \left(1 + \left(\frac{1}{\underline{\pi}(x)} - 1\right) \frac{1}{\prod_{k \neq j} \underline{\Lambda}^X_{Y_k}}\right)^{-1}.$$

4.2.5. Extremes of $\Lambda^{U_i}_X$

The case of $\Lambda^{U_i}_X$ is more complex than the others. The optimization problem cannot be created by using formula (11) in a straightforward way (or any of the subsequent rewritings), since it is constituted by the variables $\rho[x \mid u_i]$ and $\rho[x \mid \bar{u}_i]$ that are not independent. In fact, their definition in formula (19) shows that they are respectively defined on the basis of $\pi_X(u_k)$ and $\pi_X(\bar{u}_k)$ that are constrained by $\pi_X(u_k) + \pi_X(\bar{u}_k) = 1$. For this reason, initially it is used the Definition (11) of $\Lambda^{U_i}_X$, where $\rho[x \mid u_i]$ and $\rho[x \mid \bar{u}_i]$ are replaced by their original meaning given by Eq. (19). The problem of minimization is then formulated as follows,

$$\min \frac{\Lambda^X\left(\sum_{U_K \neq U_i,\ U_i = u_i} P[x \mid U] \prod_{k \neq i} \pi_X(U_k)\right) + \sum_{U_K \neq U_i,\ U_i = u_i} P[\bar{x} \mid U] \prod_{k \neq i} \pi_X(U_k)}{\Lambda^X\left(\sum_{U_K \neq U_i,\ U_i = \bar{u}_i} P[x \mid U] \prod_{k \neq i} \pi_X(U_k)\right) + \sum_{U_K \neq U_i,\ U_i = \bar{u}_i} P[\bar{x} \mid U] \prod_{k \neq i} \pi_X(U_k)},$$

$$\begin{aligned}
&\underline{\Lambda}^X \leqslant \Lambda^X \leqslant \overline{\Lambda}^X, \\
&\underline{P}[x \mid U] \leqslant P[x \mid U] \leqslant \overline{P}[x \mid U], && U \in \bigtimes_{i=1}^{n} \Omega_{U_i}, \\
&\sum_X P[X \mid U] = 1, && U \in \bigtimes_{i=1}^{n} \Omega_{U_i}, \\
&\underline{\pi}_X(u_j) \leqslant \pi_X(u_j) \leqslant \overline{\pi}_X(u_j), && j = 1, \ldots, i-1, i+1, \ldots, n, \\
&\sum_{U_j \in \{u_j, \bar{u}_j\}} \pi_X(U_j) = 1, && j = 1, \ldots, i-1, i+1, \ldots, n.
\end{aligned} \tag{21}$$

Table 1
Optimum values for the subproblem

	$\widehat{\underline{\Lambda}}_X^{U_i}(\Lambda^X)$	$\widehat{\overline{\Lambda}}_X^{U_i}(\Lambda^X)$
$\Lambda^X < 1$	$\dfrac{\overline{\widehat{\rho}}[x \mid u_i] + (\Lambda^X - 1)^{-1}}{\underline{\widehat{\rho}}[x \mid \bar{u}_i] + (\Lambda^X - 1)^{-1}}$	$\dfrac{\underline{\widehat{\rho}}[x \mid u_i] + (\Lambda^X - 1)^{-1}}{\overline{\widehat{\rho}}[x \mid \bar{u}_i] + (\Lambda^X - 1)^{-1}}$
$\Lambda^X = 1$	1	1
$\Lambda^X > 1$	$\dfrac{\underline{\widehat{\rho}}[x \mid u_i] + (\Lambda^X - 1)^{-1}}{\overline{\widehat{\rho}}[x \mid \bar{u}_i] + (\Lambda^X - 1)^{-1}}$	$\dfrac{\overline{\widehat{\rho}}[x \mid u_i] + (\Lambda^X - 1)^{-1}}{\underline{\widehat{\rho}}[x \mid \bar{u}_i] + (\Lambda^X - 1)^{-1}}$

Notice that the probability variables in problem (21), Λ^X, $P[x \mid U]$ and $\pi_X(u_K)$, are independent. Reasoning as in Section 4.2.2, the constraints of equality to 1 are removed and the new problem is such that Corollary 6 can be applied. Then the optimum is obtained at a vertex of the feasible region. This also means that, in the particular case of the $\pi_X(u_j)$ variables, the search can be restricted to the following 2^{n-1} cases: $\pi_X(u_j) \in \{\underline{\pi}_X(u_j), \overline{\pi}_X(u_j)\}$, $j \in \{1, \ldots, i-1, i+1, \ldots, n\}$. Hence, with regard to the variables $\pi_X(u_j)$, an exhaustive search is made. In other words, the solution process of problem (21) is decomposed in the above exhaustive search plus the solution of the sub-problem generated by (21) when one of the states for the $\pi_X(u_j)$ variables is fixed. Such a sub-problem can be written as follows,

$$\begin{aligned} \widehat{\underline{\Lambda}}_X^{U_i} &= \min \frac{\Lambda^X \widehat{\rho}[x \mid u_i] + (1 - \widehat{\rho}[x \mid u_i])}{\Lambda^X \widehat{\rho}[x \mid \bar{u}_i] + (1 - \widehat{\rho}[x \mid \bar{u}_i])}, \\ & \underline{\Lambda}^X \leqslant \Lambda^X \leqslant \overline{\Lambda}^X, \\ & \underline{\widehat{\rho}}[x \mid U_i] \leqslant \widehat{\rho}[x \mid U_i] \leqslant \overline{\widehat{\rho}}[x \mid U_i], \qquad U_i \in \Omega_{U_i}, \end{aligned} \tag{22}$$

where also the hat-notation is introduced. The latter is used for the quantities that depend on the chosen state of the $\pi_X(u_j)$variables (for example, $\widehat{\rho}[x \mid u_i]$ is like in Eq. (19), where the probability values $P[X \mid U]$ can vary in their respective intervals, but where the $\pi_X(u_j)$ are considered constants, because one of the following 2^{n-1} cases is fixed: $\pi_X(u_j) \in \{\underline{\pi}_X(u_j), \overline{\pi}_X(u_j)\}$, $j \in \{1, \ldots, i-1, i+1, \ldots, n\}$). Notice that now it is possible to use formula (11), since inside problem (22) the $\pi_X(u_j)$ can be considered constants, and then $\widehat{\rho}[x \mid u_i]$ and $\widehat{\rho}[x \mid \bar{u}_i]$ are not dependent. The solution of the sub-problem (22) is itself based on Corollary 6: the problem is transformed into a combinatorial one and the solution is obtained with a procedure that tests the value of the objective function when the variables take values at the two extremes of the respective intervals of definition.

A shortcut to the solution is given by first the values Λ^X can assume and then solving the remaining problem (by considering three different cases according to the value of Λ^X), like shown in Table 1 (where the objective function is rewritten according to Eq. (12)). The notation $\widehat{\underline{\Lambda}}_X^{U_i}(\Lambda^X)$ and $\widehat{\overline{\Lambda}}_X^{U_i}(\Lambda^X)$ denotes that the optimum values depend on the value chosen for Λ^X.

Notice that in order to solve problem (22), the minimum and maximum values of $\widehat{\rho}[x \mid u_i]$ and $\widehat{\rho}[x \mid \bar{u}_i]$ must be available. Such extremes are easily computed with an observation similar to that made about the extremes of $P[x \mid U]$ at the optimum; that $\underline{\widehat{\rho}}[X \mid U_i] = \sum_{U_k \neq U_i} \underline{P}[X \mid U] \prod_{k \neq i} \pi_X(U_k)$ and $\overline{\widehat{\rho}}[X \mid U_i] = \sum_{U_k \neq U_i} \overline{P}[X \mid U] \prod_{k \neq i} \pi_X(U_k)$.

This solves the minimization problem for $\Lambda_X^{U_i}$. Formula (31) summarizes the procedure for the minimization of $\Lambda_X^{U_i}$.

4.3. Interval propagation formulas

All the propagation formulas defining the algorithm are listed below:

$$\underline{P}[x \mid e] = \left(1 + \left(\frac{1}{\underline{\pi}(x)} - 1\right)\frac{1}{\underline{\Lambda}^X}\right)^{-1}, \tag{23}$$

$$\overline{P}[x \mid e] = \left(1 + \left(\frac{1}{\overline{\pi}(x)} - 1\right)\frac{1}{\overline{\Lambda}^X}\right)^{-1}, \tag{24}$$

$$\underline{\pi}(x) = \min_{\substack{j\in\{1,\ldots,n\} \\ \pi_X(u_j)\in\{\underline{\pi}_X(u_j),\overline{\pi}_X(u_j)\}}} \sum_U \underline{P}[x \mid U]\prod_i \pi_X(U_i), \tag{25}$$

$$\overline{\pi}(x) = \max_{\substack{j\in\{1,\ldots,n\} \\ \pi_X(u_j)\in\{\underline{\pi}_X(u_j),\overline{\pi}_X(u_j)\}}} \sum_U \overline{P}[x \mid U]\prod_i \pi_X(U_i), \tag{26}$$

$$\underline{\Lambda}^X = \prod_j \underline{\Lambda}_{Y_j}^X, \tag{27}$$

$$\overline{\Lambda}^X = \prod_j \overline{\Lambda}_{Y_j}^X, \tag{28}$$

$$\underline{\pi}_{Y_j}(x) = \left(1 + \left(\frac{1}{\underline{\pi}(x)} - 1\right)\frac{1}{\prod_{k\neq j}\underline{\Lambda}_{Y_k}^X}\right)^{-1}, \tag{29}$$

$$\overline{\pi}_{Y_j}(x) = \left(1 + \left(\frac{1}{\overline{\pi}(x)} - 1\right)\frac{1}{\prod_{k\neq j}\overline{\Lambda}_{Y_k}^X}\right)^{-1}, \tag{30}$$

$$\underline{\Lambda}_X^{U_i} = \min_{\substack{j\in\{1,\ldots,n\},\ j\neq i \\ \pi_X(u_j)\in\{\underline{\pi}_X(u_j),\overline{\pi}_X(u_j)\}}} \left(\min_{\Lambda^X\in\{\underline{\Lambda}^X,\overline{\Lambda}^X\}} \widehat{\underline{\Lambda}}_X^{U_i}(\Lambda^X)\right), \tag{31}$$

$$\overline{\Lambda}_X^{U_i} = \max_{\substack{j\in\{1,\ldots,n\},\ j\neq i \\ \pi_X(u_j)\in\{\underline{\pi}_X(u_j),\overline{\pi}_X(u_j)\}}} \left(\max_{\Lambda^X\in\{\underline{\Lambda}^X,\overline{\Lambda}^X\}} \widehat{\overline{\Lambda}}_X^{U_i}(\Lambda^X)\right) \tag{32}$$

where the values $\widehat{\underline{\Lambda}}_X^{U_i}(\Lambda^X)$ and $\widehat{\overline{\Lambda}}_X^{U_i}(\Lambda^X)$ are computed according to Table 1.

Formulas (23)–(32) define a distributed algorithm. The principles governing such a distributed algorithm are exactly the same as for Pearl's updating. Any node realizes a local computation. In particular, at the beginning, all the nodes are inactive, except for the nodes with a single adjacent node and the evidence nodes. An inactive node becomes active when it receives a message from one (or more) of its adjacent nodes (this is the reason why at the beginning also the evidence nodes are considered active; in fact, they are supposed to have already received a message from their dummy child). In the active state, a node is

ready to apply the opportune formulas for updating its own quantities and for computing the messages to be sent to its adjacents. An active node selects the adjacents to which it must send messages as follows:

- if the node has a single adjacent node, it must send a message to the latter;
- in the other cases, if the node received a message from just one adjacent (possibly being also the dummy child), it must send messages to the rest of its adjacents; if it received messages from more than one adjacent, it must send messages to all its adjacents (excluding the possible dummy child).

Whenever an active node has sent the proper messages to the adjacents following the above rules, it becomes inactive up to the arrival of another message.

(Observe that the above formulas and principles do not imply that an active node has immediately to send messages to its adjacents, i.e., a node may postpone this task.)

The global computation is carried out in discrete steps. At a given step, the nodes that send messages form a certain subset of the active nodes (any nonempty subset is possible); this modifies the set of active nodes for the next step; the procedure is repeated until no node is active. This condition is satisfied when any node has been updated about the global state of the network. In this state, the probability values (extremes) held by nodes are the final result of the computation.

The above description is for the *pure* application of the propagation formulas, where the calculus is fully distributed, i.e., when there is not an a-priori policy for choosing the subset of active nodes that have to send messages. But often, more efficient global computations can be made by fixing a policy depending on the supervision of the global state of the net. This is optional, and can be chosen in order to speed-up the computation. For example, Section 6 adopts this point of view. Anyway, it must be remarked that any policy gives the same result.

5. Computational complexity

The message flow is the same as in the point probability case and for this reason, its complexity is linear with the size of the network, within the class of graphs with a fixed maximum number of parents. The worst-case computation, local to a node, is determined by the calculus of an extreme of $\pi(x)$. Consider formula (25) where the minimum operator requires 2^n evaluations of the inner sum. The sum is taken over the 2^n joint states of $\{U_1, \ldots, U_n\}$. Therefore the overall complexity depends on a term $O(2^{2n_{\max}})$ in the worst case.

The term $O(2^{2n_{\max}})$ can be rewritten as $O(4^{n_{\max}})$ in order to have a more precise idea of the complexity of 2U, by making a comparison with the complexity of Pearl's updating for point probability. In fact, $O(4^{n_{\max}})$ is the local complexity of Pearl's updating for a node with $n_{\max}$ parents, where the random variables related to the node and to its parents have 4 values in their respective domains. Also notice that in the original updating the messages exchanged between nodes are 4-dimensional vectors (corresponding to the 4 values of the random variables), whereas the messages used by 2U are 2-dimensional (corresponding to the extremes of the quantity that is passed). Since the above complexity terms apply to any element of the related vector it can be stated that the local worst-case complexity of

Pearl's point probability updating (for 4-state variables) is an upper bound of 2U's local worst-case complexity.

More generally, a global computation made with 2U on a net has a worst-case complexity that is upper-bounded by the worst-case complexity of Pearl's updating on a network with the same graph but with 4-state variables.

6. A numerical example

The methodology developed above is applied to the network in Fig. 2.

For such a net, the nodes G and L represent evidence variables and it is assumed that $G = \bar{g}, L = l$. The conditional distributions of the model are listed in Table 2 (some intervals including the value 0 or the value 1 are used in order to highlight the treatment of these degenerate cases).

For simplicity of computation, instead of updating all nodes probability intervals, only node A is updated, i.e., only the extremes of $P[a \mid \bar{g}, l]$ are computed (the full application of 2U is analogous). This is done by simply choosing one among all the flows that are able to update node A about the global state of the network. The chosen flow is the following,

$$G \to D, D \to F, C \to F, F \to H, L \to H, H \to E, B \to E, E \to A$$

(recall that any flow that is able to update node A about the global state of the net is equivalent from the point of view of obtaining the correct result, as described in Section 4.3; also recall that the fully-distributed version of the algorithm could simply be implemented by making every node to act in an autonomous way, by applying its own local formulas as a consequence of any update coming from the rest of the network through one of its adjacent nodes. In the fully-distributed case, the computation would be started at the source and the barren nodes, namely A, B, C, G, L. They would send a message to their respective adjacent nodes, which would become active and would propagate the computation).

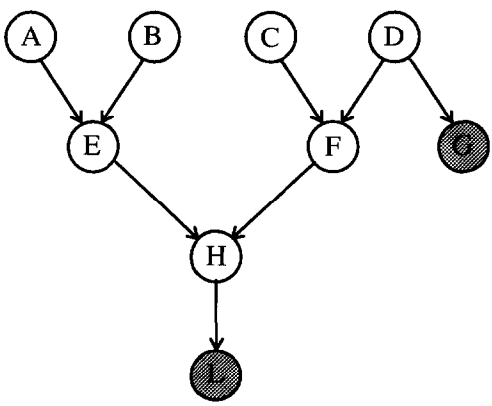

Fig. 2. An example of updating with intervals.

Table 2
Data for the application example.

Node	Conditional probabilities			
A	$P[a]$			
	[.3, .4]			
B	$P[b]$			
	[.2, .4]			
C	$P[c]$			
	[.9, 1]			
D	$P[d]$			
	[.5, .9]			
E	$P[e \mid a, b]$	$P[e \mid a, \bar{b}]$	$P[e \mid \bar{a}, b]$	$P[e \mid \bar{a}, \bar{b}]$
	[.3, .5]	[0, .2]	[.1, .3]	[.6, .7]
F	$P[f \mid c, d]$	$P[f \mid c, \bar{d}]$	$P[f \mid \bar{c}, d]$	$P[f \mid \bar{c}, \bar{d}]$
	[.1, .4]	[.5, .5]	[.5, .7]	[.8, .9]
G	$P[g \mid d]$	$P[g \mid \bar{d}]$		
	[.7, .8]	[.2, .4]		
H	$P[h \mid e, f]$	$P[h \mid e, \bar{f}]$	$P[h \mid \bar{e}, f]$	$P[h \mid \bar{e}, \bar{f}]$
	[.1, .2]	[.2, .4]	[.6, .8]	[.9, 1]
L	$P[l \mid h]$	$P[l \mid \bar{h}]$		
	[.4, .5]	[0, .2]		

The following computation is described step by step, where every step depends on the arrival of a message to a node. Such a node computes the needed quantities, based on the formulas in Section 4.3, and propagates one or more messages to others. The computation ends when node A has complete knowledge about the rest of the net. At this point, A computes the extremes of its posterior probability (Step 9).

Step 1. G $\rightarrow$ D. Node G sends the extremes of $\Lambda_{\mathrm{G}}^{\mathrm{D}}$ to D.

Formula (31) defines the message $\underline{\Lambda}_{\mathrm{G}}^{\mathrm{D}}$. The only parent of G is D, hence there does not exist a message of type π. This implies $\rho[g \mid d] = P[g \mid d]$ and $\rho[g \mid \bar{d}] = P[g \mid \bar{d}]$. Furthermore, since G is an evidence node, there exists a dummy child, $\tilde{Y}$, which sends the message $\Lambda_{\tilde{Y}}^{\mathrm{G}} = 0$ to G. By means of formulas (27) and (28), $\Lambda_{\mathrm{G}} = 0$ and formula (31) becomes, $\underline{\Lambda}_{\mathrm{G}}^{\mathrm{D}} = \widehat{\underline{\Lambda}}_{\mathrm{G}}^{\mathrm{D}}(0)$. Hence,

$$\underline{\Lambda}_{\mathrm{G}}^{\mathrm{D}} = \widehat{\underline{\Lambda}}_{\mathrm{G}}^{\mathrm{D}}(0) = \frac{\overline{P}[g \mid d] + (0-1)^{-1}}{\underline{P}[g \mid \bar{d}] + (0-1)^{-1}} = \frac{.8-1}{.2-1} = .25.$$

In the same way, by means of Eq. (32),

$$\overline{\Lambda}_{\mathrm{G}}^{\mathrm{D}} = \overline{\widehat{\Lambda}}_{\mathrm{G}}^{\mathrm{D}}(0) = \frac{\underline{P}[g \mid d] + (0-1)^{-1}}{\overline{P}[g \mid \bar{d}] + (0-1)^{-1}} = \frac{.7-1}{.4-1} = .5$$

and

$$\Lambda_{\mathrm{G}}^{\mathrm{D}} \in [.25, .5].$$

Step 2. D $\rightarrow$ F. Node D sends the extremes of $\pi_{\mathrm{F}}(d)$ to F.
D has only one child apart from F and for this reason formulas (29) and (30) become,

$$\underline{\pi}_{\mathrm{F}}(d) = \left(1 + \left(\frac{1}{\underline{\pi}(d)} - 1\right)\frac{1}{\underline{\Lambda}_{\mathrm{G}}^{\mathrm{D}}}\right)^{-1},$$

$$\overline{\pi}_{\mathrm{F}}(d) = \left(1 + \left(\frac{1}{\overline{\pi}(d)} - 1\right)\frac{1}{\overline{\Lambda}_{\mathrm{G}}^{\mathrm{D}}}\right)^{-1}.$$

Furthermore $\pi_{\mathrm{F}}(d) = P[d]$, since D is a source node. The above considerations imply,

$$\underline{\pi}_{\mathrm{F}}(d) = \left(1 + \left(\frac{1}{\underline{P}(d)} - 1\right)\frac{1}{\underline{\Lambda}_{\mathrm{G}}^{\mathrm{D}}}\right)^{-1} = \left(1 + \left(\frac{1}{.5} - 1\right)\frac{1}{.25}\right)^{-1} = .2,$$

$$\overline{\pi}_{\mathrm{F}}(d) = \left(1 + \left(\frac{1}{\overline{P}(d)} - 1\right)\frac{1}{\overline{\Lambda}_{\mathrm{G}}^{\mathrm{D}}}\right)^{-1} = \left(1 + \left(\frac{1}{.9} - 1\right)\frac{1}{.5}\right)^{-1} = .\overline{81}$$

(where $.\overline{81}$ means .818181...) and hence

$$\underline{\pi}_{\mathrm{F}}(d) \in \left[.2, .\overline{81}\right].$$

Step 3. C $\rightarrow$ F. Node C sends the extremes of $\pi_{\mathrm{F}}(c)$ to F.
In a way similar to Step 2, $\pi_{\mathrm{F}}(c) = \pi(c) = P[c]$, and

$$\pi_{\mathrm{F}}(c) \in [.9, 1].$$

Step 4. F $\rightarrow$ H. Node F sends the extremes of $\pi_{\mathrm{H}}(f)$ to H.
First, Eqs. (25) and (26) allow the extremes of $\pi(f)$ to be computed.

$$\underline{\pi}(f) = \min_{\substack{\pi_{\mathrm{F}}(c) \in \{\underline{\pi}_{\mathrm{F}}(c), \overline{\pi}_{\mathrm{F}}(c)\} \\ \pi_{\mathrm{F}}(d) \in \{\underline{\pi}_{\mathrm{F}}(d), \overline{\pi}_{\mathrm{F}}(d)\}}} \left(\sum_{\mathrm{C,D}} \underline{P}[f \mid \mathrm{C}, \mathrm{D}]\pi_{\mathrm{F}}(\mathrm{C})\pi_{\mathrm{F}}(\mathrm{D})\right),$$

$$\overline{\pi}(f) = \max_{\substack{\pi_{\mathrm{F}}(c) \in \{\underline{\pi}_{\mathrm{F}}(c), \overline{\pi}_{\mathrm{F}}(c)\} \\ \pi_{\mathrm{F}}(d) \in \{\underline{\pi}_{\mathrm{F}}(d), \overline{\pi}_{\mathrm{F}}(d)\}}} \left(\sum_{\mathrm{C,D}} \overline{P}[f \mid \mathrm{C}, \mathrm{D}]\pi_{\mathrm{F}}(\mathrm{C})\pi_{\mathrm{F}}(\mathrm{D})\right).$$

The value of the above parentheses when the state of the parents' messages is fixed are now denoted by $\underline{\widehat{\pi}}(f)$ and by $\overline{\widehat{\pi}}(f)$. The two expressions become,

$$\begin{aligned}\underline{\widehat{\pi}}(f) = {} & \underline{P}[f \mid c, d]\pi_{\mathrm{F}}(c)\pi_{\mathrm{F}}(d) + \underline{P}\left[f \mid c, \bar{d}\right]\pi_{\mathrm{F}}(c)\pi_{\mathrm{F}}\left(\bar{d}\right) \\ & + \underline{P}\left[f \mid \bar{c}, d\right]\pi_{\mathrm{F}}\left(\bar{c}\right)\pi_{\mathrm{F}}(d) + \underline{P}\left[f \mid \bar{c}, \bar{d}\right]\pi_{\mathrm{F}}\left(\bar{c}\right)\pi_{\mathrm{F}}\left(\bar{d}\right),\end{aligned}$$

$$\overline{\overline{\pi}}(f) = \overline{P}[f \mid c, d]\pi_{\mathrm{F}}(c)\pi_{\mathrm{F}}(d) + \overline{P}\big[f \mid c, \bar{d}\big]\pi_{\mathrm{F}}(c)\pi_{\mathrm{F}}\big(\bar{d}\big) + \overline{P}\big[f \mid \bar{c}, d\big]\pi_{\mathrm{F}}\big(\bar{c}\big)\pi_{\mathrm{F}}(d) + \overline{P}\big[f \mid \bar{c}, \bar{d}\big]\pi_{\mathrm{F}}\big(\bar{c}\big)\pi_{\mathrm{F}}\big(\bar{d}\big).$$

Four cases must then be considered.[3]

Case $\pi_{\mathrm{F}}(c) = \underline{\pi}_{\mathrm{F}}(c), \pi_{\mathrm{F}}(d) = \underline{\pi}_{\mathrm{F}}(d)$.

$$\underline{\widehat{\pi}}(f) = .1 \times .9 \times .2 + .5 \times .9 \times .8 + .5 \times .1 \times .2 + .8 \times .1 \times .8 = .452,$$

$$\overline{\widehat{\pi}}(f) = .4 \times .9 \times .2 + .5 \times .9 \times .8 + .7 \times .1 \times .2 + .9 \times .1 \times .8 = .518.$$

Case $\pi_{\mathrm{F}}(c) = \underline{\pi}_{\mathrm{F}}(c), \pi_{\mathrm{F}}(d) = \overline{\pi}_{\mathrm{F}}(d)$.

$$\underline{\widehat{\pi}}(f) = .1 \times .9 \times .\overline{81} + .5 \times .9 \times .\overline{18} + .5 \times .1 \times .\overline{81} + .8 \times .1 \times .\overline{18} = .21\overline{09},$$

$$\overline{\widehat{\pi}}(f) = .4 \times .9 \times .\overline{81} + .5 \times .9 \times .\overline{18} + .7 \times .1 \times .\overline{81} + .9 \times .1 \times .\overline{18} = .45.$$

Case $\pi_{\mathrm{F}}(c) = \overline{\pi}_{\mathrm{F}}(c), \pi_{\mathrm{F}}(d) = \underline{\pi}_{\mathrm{F}}(d)$.

$$\underline{\widehat{\pi}}(f) = .1 \times 1 \times .2 + .5 \times 1 \times .8 + .5 \times 0 \times .2 + .8 \times 0 \times .8 = .42,$$

$$\overline{\widehat{\pi}}(f) = .4 \times 1 \times .2 + .5 \times 1 \times .8 + .7 \times 0 \times .2 + .9 \times 0 \times .8 = .48.$$

Case $\pi_{\mathrm{F}}(c) = \overline{\pi}_{\mathrm{F}}(c), \pi_{\mathrm{F}}(d) = \overline{\pi}_{\mathrm{F}}(d)$.

$$\underline{\widehat{\pi}}(f) = .1 \times 1 \times .\overline{81} + .5 \times 1 \times .\overline{18} + .5 \times 0 \times .\overline{81} + .8 \times 0 \times .\overline{18} = .1\overline{72},$$

$$\overline{\widehat{\pi}}(f) = .4 \times 1 \times .\overline{81} + .5 \times 1 \times .\overline{18} + .7 \times 0 \times .\overline{81} + .9 \times 0 \times .\overline{18} = .4\overline{18}.$$

By taking the minimum of the $\underline{\widehat{\pi}}(f)$ and the maximum of the $\overline{\widehat{\pi}}(f)$, it follows $\pi(f) \in [.1\overline{72}, .518]$. The interval for $\pi_{\mathrm{H}}(f)$ is the same since $\pi_{\mathrm{H}}(f) = \pi(f)$ and hence,

$$\pi_{\mathrm{H}}(f) \in \big[.1\overline{72}, .518\big].$$

Step 5. L $\rightarrow$ H. Node L sends the extremes of $\Lambda_{\mathrm{L}}^{\mathrm{H}}$ to H.

H is the only parent of L, therefore $\rho[l \mid h] = P[l \mid h]$, $\rho[l \mid \bar{h}] = P[l \mid \bar{h}]$. Furthermore, L is an evidence node such that $\mathrm{L} = l$ and for this reason there exists a dummy child sending to L the message $\Lambda_{\mathrm{L}}^{\mathrm{L}} = \infty$, which implies $\Lambda^{\mathrm{L}} = \infty$. In same way as in Step 1, Eq. (31) becomes, $\underline{\Lambda}_{\mathrm{L}}^{\mathrm{H}} = \underline{\widehat{\Lambda}}_{\mathrm{L}}^{\mathrm{H}}(\infty)$, and then

$$\underline{\Lambda}_{\mathrm{L}}^{\mathrm{H}} = \underline{\widehat{\Lambda}}_{\mathrm{L}}^{\mathrm{H}}(\infty) = \frac{\underline{P}[l \mid h] + (\infty - 1)^{-1}}{\overline{P}[l \mid \bar{h}] + (\infty - 1)^{-1}} = \frac{\underline{P}[l \mid h]}{\overline{P}[l \mid \bar{h}]} = \frac{.4}{.2} = 2.$$

In an analogous way, by means of Eq. (32),

$$\overline{\Lambda}_{\mathrm{L}}^{\mathrm{H}} = \overline{\widehat{\Lambda}}_{\mathrm{L}}^{\mathrm{H}}(\infty) = \frac{\overline{P}[l \mid h] + (\infty - 1)^{-1}}{\underline{P}[l \mid \bar{h}] + (\infty - 1)^{-1}} = \frac{\overline{P}[l \mid h]}{\underline{P}[l \mid \bar{h}]} = \frac{.5}{0} = \infty.$$

The interval for $\Lambda_{\mathrm{L}}^{\mathrm{H}}$ is

$$\Lambda_{\mathrm{L}}^{\mathrm{H}} \in [2, \infty].$$

Step 6. H $\rightarrow$ E. Node H sends the extremes of $\Lambda_{\mathrm{H}}^{\mathrm{E}}$ to E.

[3] The generic message $\pi_Y(x)$ or the value $\pi(x)$ represent probabilities in x, and for this reason, $\pi_Y(\bar{x}) = 1 - \pi_Y(x)$ and $\pi(\bar{x}) = 1 - \pi(x)$.

Using formulas (31) and (32), two cases must be evaluated. These are $\pi_{\mathrm{H}}(f) = \underline{\pi}_{\mathrm{H}}(f)$ and $\pi_{\mathrm{H}}(f) = \overline{\pi}_{\mathrm{H}}(f)$. Observe that

$$\underline{\rho}[h \mid \mathrm{E}] = \sum_{\mathrm{F}} \underline{P}[h \mid \mathrm{E}, \mathrm{F}]\pi_{\mathrm{H}}(\mathrm{F}) = \underline{P}[h \mid \mathrm{E}, f]\pi_{\mathrm{H}}(f) + \underline{P}\big[h \mid \mathrm{E}, \bar{f}\,\big]\pi_{\mathrm{H}}\big(\bar{f}\,\big),$$

that

$$\overline{\rho}[h \mid \mathrm{E}] = \sum_{\mathrm{F}} \overline{P}[h \mid \mathrm{E}, \mathrm{F}]\pi_{\mathrm{H}}(\mathrm{F}) = \overline{P}[h \mid \mathrm{E}, f]\pi_{\mathrm{H}}(f) + \overline{P}\big[h \mid \mathrm{E}, \bar{f}\,\big]\pi_{\mathrm{H}}\big(\bar{f}\,\big),$$

and that $\Lambda^{\mathrm{H}} = \Lambda^{\mathrm{H}}_{\mathrm{L}}$.

Case $\pi_{\mathrm{H}}(f) = \underline{\pi}_{\mathrm{H}}(f)$.

$$\underline{\widehat{\rho}}[h \mid e] = .1 \times .1\overline{72} + .2 \times .8\overline{27} = .18\overline{27},$$

$$\overline{\widehat{\rho}}[h \mid e] = .2 \times .1\overline{72} + .4 \times .8\overline{27} = .36\overline{54},$$

$$\underline{\widehat{\rho}}[h \mid \bar{e}] = .6 \times .1\overline{72} + .9 \times .8\overline{27} = .84\overline{81},$$

$$\overline{\widehat{\rho}}[h \mid \bar{e}] = .8 \times .1\overline{72} + 1 \times .8\overline{27} = .96\overline{54}.$$

$\underline{\widehat{\Lambda}}^{\mathrm{E}}_{\mathrm{H}}(\Lambda^{\mathrm{H}})$ and $\overline{\widehat{\Lambda}}^{\mathrm{E}}_{\mathrm{H}}(\Lambda^{\mathrm{H}})$ are evaluated in the two extreme values of Λ^{H} (the approximate equality sign below denotes that the numbers are rounded).

$$\underline{\widehat{\Lambda}}^{\mathrm{E}}_{\mathrm{H}}\big(\underline{\Lambda}^{\mathrm{H}}\big) = \frac{\underline{\widehat{\rho}}[h \mid e] + (\underline{\Lambda}^{\mathrm{H}} - 1)^{-1}}{\overline{\widehat{\rho}}[h \mid \bar{e}] + (\underline{\Lambda}^{\mathrm{H}} - 1)^{-1}} = \frac{.18\overline{27} + (2-1)^{-1}}{.96\overline{54} + (2-1)^{-1}} \cong .60176,$$

$$\overline{\widehat{\Lambda}}^{\mathrm{E}}_{\mathrm{H}}\big(\underline{\Lambda}^{\mathrm{H}}\big) = \frac{\overline{\widehat{\rho}}[h \mid e] + (\underline{\Lambda}^{\mathrm{H}} - 1)^{-1}}{\underline{\widehat{\rho}}[h \mid \bar{e}] + (\underline{\Lambda}^{\mathrm{H}} - 1)^{-1}} = \frac{.36\overline{54} + (2-1)^{-1}}{.84\overline{81} + (2-1)^{-1}} \cong .73881,$$

$$\underline{\widehat{\Lambda}}^{\mathrm{E}}_{\mathrm{H}}\big(\overline{\Lambda}^{\mathrm{H}}\big) = \frac{\underline{\widehat{\rho}}[h \mid e] + (\overline{\Lambda}^{\mathrm{H}} - 1)^{-1}}{\overline{\widehat{\rho}}[h \mid \bar{e}] + (\overline{\Lambda}^{\mathrm{H}} - 1)^{-1}} = \frac{.18\overline{27} + (\infty-1)^{-1}}{.96\overline{54} + (\infty-1)^{-1}} \cong .18926,$$

$$\overline{\widehat{\Lambda}}^{\mathrm{E}}_{\mathrm{H}}\big(\overline{\Lambda}^{\mathrm{H}}\big) = \frac{\overline{\widehat{\rho}}[h \mid e] + (\overline{\Lambda}^{\mathrm{H}} - 1)^{-1}}{\underline{\widehat{\rho}}[h \mid \bar{e}] + (\overline{\Lambda}^{\mathrm{H}} - 1)^{-1}} = \frac{.36\overline{54} + (\infty-1)^{-1}}{.84\overline{81} + (\infty-1)^{-1}} \cong .43087.$$

The minimum of the two minima above corresponds to the expression in parentheses in formula (31), when the case $\pi_{\mathrm{H}}(f) = \underline{\pi}_{\mathrm{H}}(f)$ is considered. In the same way, the maximum of the two maxima corresponds to the expression in parentheses in formula (32), related to $\pi_{\mathrm{H}}(f) = \underline{\pi}_{\mathrm{H}}(f)$. Such extremes represent the temporary interval for the message, [.18926, .73881].

Case $\pi_{\mathrm{H}}(f) = \overline{\pi}_{\mathrm{H}}(f)$.

$$\underline{\widehat{\rho}}[h \mid e] = .1 \times .518 + .2 \times .482 = .1482,$$

$$\overline{\widehat{\rho}}[h \mid e] = .2 \times .518 + .4 \times .482 = .2964,$$

$$\underline{\widehat{\rho}}[h \mid \bar{e}] = .6 \times .518 + .9 \times .482 = .84\overline{81},$$

$$\overline{\widehat{\rho}}[h \mid \bar{e}] = .8 \times .518 + 1 \times .482 = .8964.$$

As above, $\underline{\widehat{\Lambda}}^{\mathrm{E}}_{\mathrm{H}}(\Lambda^{\mathrm{H}})$ and $\overline{\widehat{\Lambda}}^{\mathrm{E}}_{\mathrm{H}}(\Lambda^{\mathrm{H}})$ are computed according to the two extreme values of Λ^{H},

$$\underline{\widehat{\Lambda}}^{\mathrm{E}}_{\mathrm{H}}(\underline{\Lambda}^{\mathrm{H}}) = \frac{\underline{\widehat{\rho}}[h \mid e] + (\underline{\Lambda}^{\mathrm{H}} - 1)^{-1}}{\overline{\widehat{\rho}}[h \mid \bar{e}] + (\underline{\Lambda}^{\mathrm{H}} - 1)^{-1}} = \frac{.1482 + (2-1)^{-1}}{.8964 + (2-1)^{-1}} \cong .60546,$$

$$\overline{\widehat{\Lambda}}^{\mathrm{E}}_{\mathrm{H}}(\underline{\Lambda}^{\mathrm{H}}) = \frac{\overline{\widehat{\rho}}[h \mid e] + (\underline{\Lambda}^{\mathrm{H}} - 1)^{-1}}{\underline{\widehat{\rho}}[h \mid \bar{e}] + (\underline{\Lambda}^{\mathrm{H}} - 1)^{-1}} = \frac{.2964 + (2-1)^{-1}}{.7446 + (2-1)^{-1}} \cong .74309,$$

$$\underline{\widehat{\Lambda}}^{\mathrm{E}}_{\mathrm{H}}(\overline{\Lambda}^{\mathrm{H}}) = \frac{\underline{\widehat{\rho}}[h \mid e] + (\overline{\Lambda}^{\mathrm{H}} - 1)^{-1}}{\overline{\widehat{\rho}}[h \mid \bar{e}] + (\overline{\Lambda}^{\mathrm{H}} - 1)^{-1}} = \frac{.1482 + (\infty-1)^{-1}}{.8964 + (\infty-1)^{-1}} \cong .16533,$$

$$\overline{\widehat{\Lambda}}^{\mathrm{E}}_{\mathrm{H}}(\overline{\Lambda}^{\mathrm{H}}) = \frac{\overline{\widehat{\rho}}[h \mid e] + (\overline{\Lambda}^{\mathrm{H}} - 1)^{-1}}{\underline{\widehat{\rho}}[h \mid \bar{e}] + (\overline{\Lambda}^{\mathrm{H}} - 1)^{-1}} = \frac{.2964 + (\infty-1)^{-1}}{.7446 + (\infty-1)^{-1}} \cong .39807.$$

Like in the previous case, the four values are used to identify a temporary interval for the message, [.16533, .74309], whose extremes, respectively, correspond to the values in parentheses in formulas (31) and (32), when the case $\pi_{\mathrm{H}}(f) = \overline{\pi}_{\mathrm{H}}(f)$ is taken into account.

The interval for the message is obtained by applying the outer minimum operator of formula (31) and the outer maximum operator of formula (32), i.e., choosing the minimum between the left extremes of the temporary intervals and the maximum of the right extremes.

$$\Lambda^{\mathrm{E}}_{\mathrm{H}} \in [.16533, .74309].$$

Step 7. B $\rightarrow$ E. Node B sends the extremes of $\pi_{\mathrm{E}}(b)$ to E.

The result is straightforward, since $\pi_{\mathrm{E}}(b) = \pi(b) = P[b]$.

$$\pi_{\mathrm{E}}(b) \in [.2, .4].$$

Step 8. E $\rightarrow$ A. Node E sends the extremes of $\Lambda^{\mathrm{A}}_{\mathrm{E}}$ to A.

The procedure is similar to Step 6. Initially two cases must be disposed, $\pi_{\mathrm{E}}(b) = \underline{\pi}_{\mathrm{E}}(b)$ and $\pi_{\mathrm{E}}(b) = \overline{\pi}_{\mathrm{E}}(b)$, by using

$$\underline{\rho}[e \mid \mathrm{A}] = \sum_{\mathrm{B}} \underline{P}[e \mid \mathrm{A}, \mathrm{B}]\pi_{\mathrm{E}}(\mathrm{B}) = \underline{P}[e \mid \mathrm{A}, b]\pi_{\mathrm{E}}(b) + \underline{P}\big[e \mid \mathrm{A}, \bar{b}\big]\pi_{\mathrm{E}}\big(\bar{b}\big),$$

$$\overline{\rho}[e \mid \mathrm{A}] = \sum_{\mathrm{B}} \overline{P}[e \mid \mathrm{A}, \mathrm{B}]\pi_{\mathrm{E}}(\mathrm{B}) = \overline{P}[e \mid \mathrm{A}, b]\pi_{\mathrm{E}}(b) + \overline{P}\big[e \mid \mathrm{A}, \bar{b}\big]\pi_{\mathrm{E}}\big(\bar{b}\big)$$

and $\Lambda^{\mathrm{E}} = \Lambda^{\mathrm{E}}_{\mathrm{H}}$.

Case $\pi_{\mathrm{E}}(b) = \underline{\pi}_{\mathrm{E}}(b)$. It is,

$$\underline{\widehat{\rho}}[e \mid a] = .3 \times .2 + 0 \times .8 = .06, \quad \overline{\widehat{\rho}}[e \mid a] = .5 \times .2 + .2 \times .8 = .26,$$

$$\underline{\widehat{\rho}}[e \mid \bar{a}] = .1 \times .2 + .6 \times .8 = .5, \quad \overline{\widehat{\rho}}[e \mid \bar{a}] = .3 \times .2 + .7 \times .8 = .62.$$

$\underline{\widehat{\Lambda}}^{\mathrm{A}}_{\mathrm{E}}(\Lambda^{\mathrm{E}})$ and $\overline{\widehat{\Lambda}}^{\mathrm{A}}_{\mathrm{E}}(\Lambda^{\mathrm{E}})$ are evaluated in the two extreme values of Λ^{E}.

$$\underline{\widehat{\Lambda}}^{\mathrm{A}}_{\mathrm{E}}(\underline{\Lambda}^{\mathrm{E}}) = \frac{\overline{\widehat{\rho}}[e \mid a] + (\underline{\Lambda}^{\mathrm{E}} - 1)^{-1}}{\underline{\widehat{\rho}}[e \mid \bar{a}] + (\underline{\Lambda}^{\mathrm{E}} - 1)^{-1}} \cong \frac{.26 + (.16533 - 1)^{-1}}{.5 + (.16533 - 1)^{-1}} \cong 1.3438,$$

$$\overline{\overline{\Lambda}}{}^{\mathrm{A}}_{\mathrm{E}}(\underline{\Lambda}^{\mathrm{E}}) = \frac{\widehat{\underline{\rho}}[e \mid a] + (\underline{\Lambda}^{\mathrm{E}} - 1)^{-1}}{\overline{\widehat{\rho}}[e \mid \bar{a}] + (\underline{\Lambda}^{\mathrm{E}} - 1)^{-1}} \cong \frac{.06 + (.16533 - 1)^{-1}}{.62 + (.16533 - 1)^{-1}} \cong 1.9687,$$

$$\underline{\widehat{\Lambda}}{}^{\mathrm{A}}_{\mathrm{E}}(\overline{\Lambda}^{\mathrm{E}}) = \frac{\overline{\widehat{\rho}}[e \mid a] + (\overline{\Lambda}^{\mathrm{E}} - 1)^{-1}}{\underline{\widehat{\rho}}[e \mid \bar{a}] + (\overline{\Lambda}^{\mathrm{E}} - 1)^{-1}} \cong \frac{.26 + (.74309 - 1)^{-1}}{.5 + (.74309 - 1)^{-1}} \cong 1.0707,$$

$$\overline{\overline{\Lambda}}{}^{\mathrm{A}}_{\mathrm{E}}(\overline{\Lambda}^{\mathrm{E}}) = \frac{\widehat{\underline{\rho}}[e \mid a] + (\overline{\Lambda}^{\mathrm{E}} - 1)^{-1}}{\overline{\widehat{\rho}}[e \mid \bar{a}] + (\overline{\Lambda}^{\mathrm{E}} - 1)^{-1}} \cong \frac{.06 + (.74309 - 1)^{-1}}{.62 + (.74309 - 1)^{-1}} \cong 1.1711.$$

The first temporary interval is [.10707, 1.9687].

Case $\pi_{\mathrm{E}}(b) = \overline{\pi}_{\mathrm{E}}(b)$ implies,

$$\underline{\widehat{\rho}}[e \mid a] = .3 \times .4 + 0 \times .6 = .12, \quad \overline{\widehat{\rho}}[e \mid a] = .5 \times .4 + .2 \times .6 = .32,$$

$$\underline{\widehat{\rho}}[e \mid \bar{a}] = .1 \times .4 + .6 \times .6 = .4, \quad \overline{\widehat{\rho}}[e \mid \bar{a}] = .3 \times .4 + .7 \times .6 = .54.$$

$\underline{\widehat{\Lambda}}{}^{\mathrm{A}}_{\mathrm{E}}(\Lambda^{\mathrm{E}})$ and $\overline{\overline{\Lambda}}{}^{\mathrm{A}}_{\mathrm{E}}(\Lambda^{\mathrm{E}})$ become,

$$\underline{\widehat{\Lambda}}{}^{\mathrm{A}}_{\mathrm{E}}(\underline{\Lambda}^{\mathrm{E}}) = \frac{\overline{\widehat{\rho}}[e \mid a] + (\underline{\Lambda}^{\mathrm{E}} - 1)^{-1}}{\underline{\widehat{\rho}}[e \mid \bar{a}] + (\underline{\Lambda}^{\mathrm{E}} - 1)^{-1}} \cong \frac{.32 + (.16533 - 1)^{-1}}{.4 + (.16533 - 1)^{-1}} \cong 1.1002,$$

$$\overline{\overline{\Lambda}}{}^{\mathrm{A}}_{\mathrm{E}}(\underline{\Lambda}^{\mathrm{E}}) = \frac{\widehat{\underline{\rho}}[e \mid a] + (\underline{\Lambda}^{\mathrm{E}} - 1)^{-1}}{\overline{\widehat{\rho}}[e \mid \bar{a}] + (\underline{\Lambda}^{\mathrm{E}} - 1)^{-1}} \cong \frac{.12 + (.16533 - 1)^{-1}}{.54 + (.16533 - 1)^{-1}} \cong 1.6382,$$

$$\underline{\widehat{\Lambda}}{}^{\mathrm{A}}_{\mathrm{E}}(\overline{\Lambda}^{\mathrm{E}}) = \frac{\overline{\widehat{\rho}}[e \mid a] + (\overline{\Lambda}^{\mathrm{E}} - 1)^{-1}}{\widehat{\underline{\rho}}[e \mid \bar{a}] + (\overline{\Lambda}^{\mathrm{E}} - 1)^{-1}} \cong \frac{.32 + (.74309 - 1)^{-1}}{.4 + (.74309 - 1)^{-1}} \cong 1.0229,$$

$$\overline{\overline{\Lambda}}{}^{\mathrm{A}}_{\mathrm{E}}(\overline{\Lambda}^{\mathrm{E}}) = \frac{\widehat{\underline{\rho}}[e \mid a] + (\overline{\Lambda}^{\mathrm{E}} - 1)^{-1}}{\overline{\widehat{\rho}}[e \mid \bar{a}] + (\overline{\Lambda}^{\mathrm{E}} - 1)^{-1}} \cong \frac{.12 + (.74309 - 1)^{-1}}{.54 + (.74309 - 1)^{-1}} \cong 1.1253.$$

The second temporary interval is [.10229, 1.6382] and then the interval for the message is

$$\Lambda^{\mathrm{A}}_{\mathrm{E}} \in [1.0229, 1.9687].$$

Step 9. Node A computes the interval of $P[a \mid \bar{g}, l]$.

Formulas (23) and (24) become,

$$\underline{P}[a \mid \bar{g}, l] = \left(1 + \left(\frac{1}{\underline{\pi}(a)} - 1\right)\frac{1}{\underline{\Lambda}^{\mathrm{A}}}\right)^{-1}$$

and

$$\overline{P}[a \mid \bar{g}, l] = \left(1 + \left(\frac{1}{\overline{\pi}(a)} - 1\right)\frac{1}{\overline{\Lambda}^{\mathrm{A}}}\right)^{-1},$$

since $\pi(a) = P[a]$ and $\Lambda^{\mathrm{A}} = \Lambda_{\mathrm{E}}^{\mathrm{A}}$.

$$\underline{P}[a \mid \bar{g}, l] = \left(1 + \left(\frac{1}{\underline{\pi}(a)} - 1\right)\frac{1}{\underline{\Lambda}^{\mathrm{A}}}\right)^{-1} \cong \left(1 + \left(\frac{1}{.3} - 1\right)\frac{1}{1.0229}\right)^{-1} \cong .30478,$$

$$\overline{P}[a \mid \bar{g}, l] = \left(1 + \left(\frac{1}{\overline{\pi}(a)} - 1\right)\frac{1}{\overline{\Lambda}^{\mathrm{A}}}\right)^{-1} \cong \left(1 + \left(\frac{1}{.4} - 1\right)\frac{1}{1.9687}\right)^{-1} \cong .56756,$$

and hence

$$P[a \mid \bar{g}, l] \in [.30478, .56756].$$

7. Discussion

This paper focuses on the treatment of credal sets over Bayesian networks. Credal sets are a general theory for dealing with uncertainty, with meaningful interpretation, and able to relax the precision requirement of Bayesian theory [16]. Bayesian networks are a widely-used tool to structure and solve complex reasoning problems. Their union seems a very flexible way of modeling knowledge.

Unfortunately, the flexibility provided to the user generates complex problems to be dealt with in order to develop a reasoning system. Such problems can be seen as global optimization problems of nonlinear functions over polytopes. It is clear that there is the need of a remarkable effort in order to provide effective solutions to the problems above. To date, it is unknown whether a general effective solution algorithm can be realized or if, for example, a more promising direction is the solution of less general instances of the problem.

The reasons for the relatively slow steps in this sector are different.

- One of them may be a partial unawareness about credal sets. In fact, especially their subset of probability intervals may have been misleading, bringing to the wrong concept that intervals are a straight generalization of probability. This is related to the idea that there is one unknown distribution in the credal set that is the real one, and that the credal set is a tool for making sensitivity analysis. We would like to point out, like Walley does [16], that credal sets are a *new* theory, that of course generalizes probability, but with its own characteristics, thus *not being a simple extension*. The same relevant connection of credal sets with the optimization world seems to be a conceptual jump that only recently is becoming clearer.
- Another reason may be the impossibility to make experiments with credal networks, from which to get new ideas for research. In fact, the absence of an effective algorithm for a significant set of networks may have limited the interest and the study of the subject.

The two main contributions of this paper have been originated by the need of dealing with the considerations above. One of them is the precise definition of credal network and of the related optimization problems and their properties. This should constitute a basis clarifying what is a credal network and what are the type of problems to be dealt with when

working with credal networks. This is provided by the formalization given in Section 2 and in Appendix A. On its basis, the paper provides the results below.

- The functions of type-3 are defined, thus stating a general definition of many quantities which depend on the probability distribution over the network. Such a definition shows that the nature of the problems is nonlinear. This has to be taken into account when looking for solution procedures. In fact, the problem of local minima becomes predominant.
- The combinatorial nature of the optimization problems is shown. Such a result is general, being true for any type of Bayesian network with convex sets of probability. It states that a function of type-3 has optima at the vertices of the polytope of definition. This constitutes a general tool that can be exploited in order to develop solution procedures. It also suggests that pure combinatorial solution procedures should be avoided, since the number of dimensions can be extremely great, and then the number of vertices of the polytope can be huge.

The second contribution of the present paper is the development of 2U. 2U is, in our knowledge, the first exact linear-time algorithm for the updating of a wide set of credal nets like the singly-connected nets with binary random variables. The updating process is carried out considering the distributions compatible with the intervals, i.e., all the distributions in $\wp$. The updating can be interpreted as a distributed optimization process. In a time which is linear to the size of the network, 2U solves $2N$ optimization problems, with a message passing scheme similar to Pearl's updating.

It remains to establish whether a more general effective exact algorithm exists. A straight generalization of 2U and also the development of a general distributed algorithm does not seem easy. 2U is successful because it relies on the independence of messages and on a controlled growth of the sets to be treated. Specifically, the following observations are made.

- The use of a singly-connected net ensures that messages from different parts of the net are independent. When working with multiply-connected nets, the above characteristic is lost. In the Bayesian network literature, two main ways are used to turn a multiply-connected net into a singly-connected net, i.e., clustering and conditioning (see the work of Pearl for an introduction [12]). Conditioning works by instantiating a set of nodes (called loop-cutset) that allow the loops to be opened (in the example of Fig. 3, node A can be instantiated for this purpose, and the arc from A to C can then be removed, thus obtaining the net in Fig. 4). Then, the computation is carried out on the resulting singly-connected net as many times as many joint states of the variables in the cutset exist, and the results are summed in order to obtain the final result related to the multiply-connected net. For example, consider the computation of $P[\mathrm{D}]$ on the multiply-connected net in Fig. 3. The cutset is $\{\mathrm{A}\}$. Assuming A is binary, two computations are executed on the resulting singly-connected net (Fig. 4), which are related to $\mathrm{A} = a$ and $\mathrm{A} = \bar{a}$. The application of Pearl's formulas, respectively, produces $P[\mathrm{D}, \mathrm{A} = a]$ and $P[\mathrm{D}, \mathrm{A} = \bar{a}]$. Their sum is just $P[\mathrm{D}]$. This way of solving multiply-connected networks does not seem promising for credal sets. Consider the case when the net in Fig. 3 is a credal network and the computation of the minimum of $P[\mathrm{D}]$ must be realized. Assume that $\underline{P}[\mathrm{D}, \mathrm{A} = a]$ and $\underline{P}[\mathrm{D}, \mathrm{A} = \bar{a}]$ can be computed. Now, in order to obtain $\underline{P}[\mathrm{D}]$, it is not possible

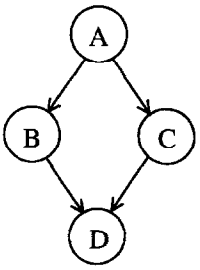

Fig. 3. A multiply-connected net.

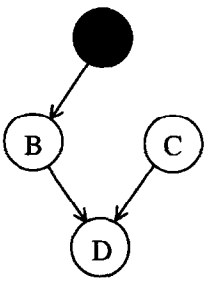

Fig. 4. The application of conditioning.

Fig. 5. The cluster net.

to simply make $\underline{P}[\mathrm{D}, \mathrm{A} = a] + \underline{P}[\mathrm{D}, \mathrm{A} = \bar{a}]$. In fact the minimum of $P[\mathrm{D}, \mathrm{A} = a]$ is generally related to a joint distribution $P'[\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}]$ that is different from the joint distribution $P''[\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}]$ from which the minimum of $P[\mathrm{D}, \mathrm{A} = \bar{a}]$ is attained. Of course, summing probability values from different distributions (in $\wp$) does not make sense. Therefore, the application of conditioning should rely on the constraint that the joint distribution is the same in both the computations, i.e., that $P' = P''$. Such a *global* constraint across the two problems (executions on the singly-connected net) seems difficult to require, just for its nonlocal nature. Avoiding global requirements is possible using clustering in the place of conditioning, like explained in the following point.

- In order to turn the net in Fig. 3 to a singly-connected credal net, the nodes B and C can be clustered into a single node (like in Fig. 5). The credal sets for the posterior distributions of the new node, BC, is computed by a simple combinatorial procedure. The latter generates the credal sets for $P[\mathrm{B}, \mathrm{C} \mid \mathrm{A}]$ by multiplying the extreme points of the credal sets for $P[\mathrm{B} \mid \mathrm{A}]$ and $P[\mathrm{C} \mid \mathrm{A}]$, and by taking the convex hull of the resulting sets [5]. The above observations suggest that the problem of multiple connection might be treated by passing to singly-connected cluster nets. Hence the problem seems to shift to the use of multistate variables.
- The use of discrete multistate variables is a jump to an n-dimensional space that makes the problem harder. In fact, the use of binary variables allows the computation to be carried out only passing intervals between nodes (see the derivation of 2U in

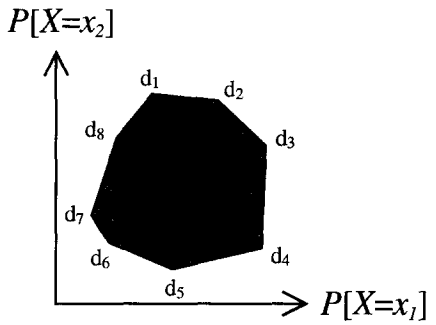

Fig. 6. The region for the distribution of X.

Section 4.2). This means that the quantities that are exchanged during the computation always need only two numbers to be described (this is the case of the controlled growth). On the contrary, moving to at least 2 dimensions (that is, to random variables with at least 3 states) implies that the sets manipulated for the updating require an a-priori unbounded number of vertices to be described (therefore a possibly uncontrolled growth is faced), like the following argument illustrates. At a given stage of the propagation, for example, the definition set of the prior probability distribution of a 3-state variable $X \in \{x_1, x_2, x_3\}$ might be like in Fig. 6. Such a set is obtained by substituting $P[X = x_3]$ with $1 - P[X = x_1] - P[X = x_2]$ and by drawing the plan region determined by the constraints on $P[X = x_1]$, $P[X = x_2]$ and $P[X = x_3]$. The region in Fig. 6 can also be seen as the convex hull of the extreme distributions corresponding to the vertices denoted by $d_i, i = 1, \ldots, 8$. Hence, just 8 points are needed in order to exactly describe the region. Suppose that such a region is used for the purpose of computing the distribution of another variable. In a graphical model, this usually passes through the operations of multiplication and marginalization. The multiplication of the extreme points of different regions produces some joint probability values. The marginalization drops from the join probability values the unwanted random variables. Suppose that the multiplication is made using two regions of 8 extreme points each. Since all the combinations must be taken into account, the number of points generated by the product is 8^2. In general, *not* all such points are vertices of the resulting region [5], i.e., some of them can be inner points. Furthermore, the marginalization can again reduce the number of such extreme points, producing a final region that is described with less than 8^2 extreme distributions. Anyway, the intermediate (multiplication) step making the number of points be exponentially increased, seems to favour a possible uncontrollable growth. That is, it seems probable that the newly generated region has to be described in greater detail than that used for the ones that originated it. This would imply that the new regions for the distributions that are created during the inference might require a growing number of points to be described, thus also requiring a growing computational time to take all these points into account during the computation. Currently, it is unknown whether there exists some type of regularity

in the growth that can be exploited in order to have a synthetic description of such sets.

The points just described evidence the need of further efforts in order to structure the present field and to study closer the nature of the problems. Some possible research directions include: the use of pure optimization methods, like descent algorithms or the generalized geometric programming approach [17]; the development of approximate algorithms; the definition of the lines within which the treated problems are computationally viable.

Acknowledgement

The authors would like to thank two anonymous referees for their precious comments.

Appendix A

The following lemma is used in the proof of Theorem 5.

Lemma A.1. *A ratio of two linear functions defined over a polytope has optima at the vertices of the feasible set.*

Proof. Consider the function $l(x) = h(x)/g(x)$ defined on a polytope $\widetilde{\Omega}$, where $h(x)$ and $g(x)$ are linear. Define the function $l_\theta(x) = h(x) - \theta g(x)$, with parameter $\theta \in \mathbb{R}$. It is a known result in fractional programming [10] that there always exists a value $\theta^* \in \mathbb{R}$ such that if x^* is a global optimum point for l_{θ^*}, it is a global optimum point for l too. Since l_{θ^*} is linear, x^* must be found at a vertex of $\widetilde{\Omega}$. □

Proof of Theorem 5. Notice that since the variables of different sets of the partition vary in the different regions Ω_j, if $x_{\mathfrak{I}}$ is a vertex of Ω, then $x_{\mathfrak{I}_j}$ is a vertex of Ω_j $\forall j = 1, \ldots, k$.

Consider the case of a function f of type-1. f has extremes because it is a continuous function defined on a closed and bounded set. By contradiction, let $x^*_{\mathfrak{I}} = \{x^*_{\mathfrak{I}_1}, \ldots, x^*_{\mathfrak{I}_k}\}$ be a global extreme point that is not a vertex of Ω. For the initial observation, there must exist at least one $j \in \{1, \ldots, k\}$ such that $x^*_{\mathfrak{I}_j}$ is not a vertex in Ω_j. Consider the function $f_{\mathfrak{I}\setminus\mathfrak{I}_j}$ obtained from f by fixing the values of the variables indexed by $\mathfrak{I} \setminus \mathfrak{I}_j$, according to $x^*_{\mathfrak{I}}$. $f_{\mathfrak{I}\setminus\mathfrak{I}_j}$ is linear by definition, but by the fundamental theorem of linear programming, a linear function has extremes at the vertices of the feasible polytope. In this case, there must exist a point $x^{**}_{\mathfrak{I}_j}$, a vertex of Ω_j, which is a global extremum point of $f_{\mathfrak{I}\setminus\mathfrak{I}_j}$ in Ω_j. This is a contradiction since, letting $x^\star_{\mathfrak{I}} = \{x^*_{\mathfrak{I}_1}, \ldots, x^*_{\mathfrak{I}_{j-1}}, x^{**}_{\mathfrak{I}_j}, x^*_{\mathfrak{I}_{j+1}}, \ldots, x^*_{\mathfrak{I}_k}\}$, $f(x^\star_{\mathfrak{I}})$ should strictly improve $f(x^*_{\mathfrak{I}})$.

Consider a type-2 function. The proof follows the same line as for functions of type-1, but in this case, $f_{\mathfrak{I}\setminus\mathfrak{I}_j}$ is the ratio of two linear functions. The proof follows on the basis of Lemma A.1. □

Proof of Theorem 7. Let $N = \{1, \ldots, n\}$ denote the set of nodes of the graph.

Initially, consider the case when ξ is a prior probability. The Factorization Theorem [12] states that

$$P[X_N] = \varsigma_{X_N} = \prod_{i \in N} \varsigma_{i, X_i^{\downarrow N}}^{X_{Pa(i)}^{\downarrow N}}. \tag{A.1}$$

By definition, any prior probability is the sum of terms of the form given by the right side of Eq. (A.1),

$$\xi = \sum_{X_N \in \widetilde{\Omega}} \prod_{i \in N} \varsigma_{i, X_i^{\downarrow N}}^{X_{Pa(i)}^{\downarrow N}} \tag{A.2}$$

for a certain $\widetilde{\Omega} \subseteq \Omega_N$. In the case of a credal network, ξ is a function of the variables $\varsigma_{i, X_i^{\downarrow N}}^{X_{Pa(i)}^{\downarrow N}}$.

In addition, ξ is a type-3 function, in fact: Eq. (A.2) shows that ξ is of type-1; the partition of $\Im$ is the collection of sets $\Im_i^{Pa(i)} = \{(i, X_i, X_{Pa(i)}) \mid X_i \in \Omega_i\}$, $i \in N$, $X_{Pa(i)} \in \Omega_{Pa(i)}$; finally, $\xi_{\Im \setminus \Im_j^{Pa(j)}}$ is a linear function $\forall j \in N$, $\forall X_{Pa(j)} \in \Omega_{Pa(j)}$. In fact, any term in the sum of Eq. (A.2) contains exactly one variable for every node, therefore any term in the sum can contain at most one variable indexed by $\Im_j^{Pa(j)}$.

The computation of an extreme of ξ is formalized by constraining ξ on the domain where every set of variables indexed by $\Im_i^{Pa(i)}$ varies in its own polytope $\wp_i^{X_{Pa(i)}}$. This problem satisfies the hypotheses of Theorem 5. The case of posterior probabilities is handled in a similar way, the only difference is that a posterior probability is the ratio of two terms of the form (A.2), and is therefore a type-2 function. □

References

[1] L. Campos, J. Huete, Independence concepts in upper and lower probabilities, in: B. Bouchon-Meunier, L. Valverde, R.R. Yager (Eds.), Uncertainty in Intelligent Systems, Elsevier, Amsterdam, 1993, pp. 85–96.

[2] L. Campos, J. Huete, S. Moral, Probability intervals: a tool for uncertain reasoning, Internat. J. of Uncertainty, Fuzziness and Knowledge-Based Systems 2 (2) (1994) 167–196.

[3] L. Campos, S. Moral, Propagating uncertain information forward, Internat. J. Intell. Syst. 7 (1992) 15–24.

[4] L. Campos, S. Moral, Independence concepts for convex sets of probabilities, in: P. Besnard, S. Hanks (Eds.), Proceedings 11th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Mateo, CA, 1995, pp. 108–115.

[5] J.E. Cano, S. Moral, J.F. Verdegay-Lòpez, Propagation of convex sets of probabilities in directed acyclic networks, in: B. Bouchon-Meunier, L. Valverde, R.R. Yager (Eds.), Uncertainty in Intelligent Systems, Elsevier, Amsterdam, 1993, pp. 15–26.

[6] L. Chrisman, Propagation of 2-monotone lower probabilities on an undirected graph, in: E. Horvitz, F. Jensen (Eds.), Proceedings 12th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, 1996, pp. 178–185.

[7] L. Chrisman, Independence with lower and upper probabilities, in: E. Horvitz, F. Jensen (Eds.), Proceedings 12th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, 1996, pp. 169–177.

[8] F. Cozman, A brief introduction to quasi-Bayesian theory (and lower probability, lower expectations, Choquet capacities, robust Bayesian methods, etc.) for AI, tutorial paper, http://www.auai.org/auai-tutes.html, 1996.

[9] F. Cozman, Robustness analysis of Bayesian networks with finitely generated convex-sets of distributions, Technical Report No. CMU-RI-TR 96-41, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1996.
[10] B.D. Craven, Fractional Programming, Heldermann, Berlin, 1988.
[11] K.W. Fertig, J.S. Breese, Interval influence diagrams, in: M. Henrion, R.D. Shachter, L.N. Kanal, J.F. Lemmer (Eds.), Uncertainty in Artificial Intelligence 5, North-Holland, Amsterdam, 1990, pp. 149–161.
[12] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, CA, 1988.
[13] B. Tessem, Interval probability propagation, Internat. J. Approx. Reasoning 7 (3) (1992) 95–120.
[14] S.A. Vavasis, Complexity issues in global optimization: a survey, in: R. Horst, P.M. Pardalos (Eds.), Handbook of Global Optimization, Kluwer, Dordrecht, The Netherlands, 1995, pp. 27–41.
[15] P. Walley, Statistical Reasoning with Imprecise Probabilities, Chapman and Hall, New York, 1991.
[16] P. Walley, Measures of uncertainty in expert systems, Artificial Intelligence 83 (1996) 1–58.
[17] M. Zaffalon, Inferenze e Decisioni in Condizioni di Incertezza con Modelli Grafici Orientati, Ph.D. Thesis, Matematica Computazionale e Ricerca Operativa, Università degli Studi di Milano, Italy, 1997.